

TUGAS AKHIR - KI141502

PERMAINAN FIRST PERSON SHOOTER (FPS) REALITAS VIRTUAL PERANGKAT ANDROID SURVIVAL MAZE

**GURUH ARYA SENNA
NRP 5113100010**

**Dosen Pembimbing
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.
Imam Kuswardayan, S.Kom., M.T.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

PERMAINAN FIRST PERSON SHOOTER (FPS) REALITAS VIRTUAL PERANGKAT ANDROID SURVIVAL MAZE

**GURUH ARYA SENNA
NRP 5113100010**

**Dosen Pembimbing
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.
Imam Kuswardayan, S.Kom., M.T.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman Ini Sengaja Dikosongkan]



FINAL PROJECT- KI141502

FIRST PERSON SHOOTER (FPS) GAME SURVIVAL MAZE VIRTUAL REALITY FOR ANDROID DEVICE

**GURUH ARYA SENNA
NRP 5113100010**

**Advisor
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.
Imam Kuswardayan, S.Kom., M.T.**

**DEPARTMENT OF INFORMATICS
Faculty of Information and Communication Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PERMAINAN *FIRST PERSON SHOOTER* (FPS) REALITAS VIRTUAL PERANGKAT ANDROID *SURVIVAL MAZE*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

GURUH ARYA SENNA

NRP. 5113100010

Disetujui oleh Dosen Pembimbing 1)
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.
NIP: 197712172003121001

Imam Kuswardayan, S.Kom., M.
NIP: 197612152003121001



**SURABAYA
JANUARI, 2018**

[Halaman ini sengaja dikosongkan]

**PERMAINAN *FIRST PERSON SHOOTER* (FPS)
REALITAS VIRTUAL PERANGKAT ANDROID
*SURVIVAL MAZE***

Nama Mahasiswa : Guruh Arya Senna
NRP : 5113100010
Departemen : Informatika FTIK-ITS
Dosen Pembimbing I : Dr.Eng. Darlis Herumurti,
S.Kom., M.Kom.
Dosen Pembimbing II : Imam Kuswardayan, S.Kom.,
M.T.

Abstrak

Game merupakan salah satu media hiburan yang sedang pesat-pesatnya berkembang dan bermain game adalah salah satu cara untuk mengisi waktu luang. Hingga saat ini banyak sekali jenis game yang sudah ada, selain itu semakin bertambah pula fitur pembangun maupun inovasi pendukung untuk suatu game. Game yang hingga kini berkembang diantaranya game bergenre first person shooter (FPS).

Fitur-fitur pembangun game selalu ditingkatkan agar semakin menarik perhatian dan membuat pemain/gamer lebih betah dalam bermain, diantaranya berkembang fitur Non Playable Character (NPC). NPC memiliki fungsi esensial dalam artifisial dunia game sehingga lebih immersive, terdiri beragam macam, salah satunya yang sangat menambah tantangan saat bermain yaitu karakter musuh. Selain itu, saat ini sedang berkembang inovasi-inovasi baru pendukung sebuah game, terutama untuk game pada komputer maupun perangkat bergerak, salah satunya adalah teknologi bernama virtual reality/realitas virtual.

‘Survival Maze’ merupakan inovasi FPS yang dirancang berbasis Realitas Virtual Perangkat Android. Sebagaimana aplikasi realitas virtual lainnya, game ini akan menggunakan

sudut pandang orang pertama, ditambah terdapat karakter musuh agar game lebih menarik dan menantang bagi pemain. Karakter musuh nantinya akan dilengkapi dengan metode Finite State Machine (FSM) dalam pembuatan perilakunya.

Pengujian aplikasi dilakukan dengan mengoperasikannya pada perangkat berbeda dan juga melalui kuesioner responden yang pernah memainkan permainan sejenis, yaitu jenis FPS ataupun jenis maze/labirin sebelumnya. Berdasarkan pengujian, aplikasi permainan ini dapat menampilkan objek-objek dalam tampilan realitas virtual serta tidak ada kebutuhan fungsional yang tidak bekerja untuk fitur-fitur dasar yang diperlukan pada aplikasi permainan ini. Dengan dikembangkannya aplikasi ini, diharapkan dapat menjadi alternatif game dengan jenis serupa dan menjadi dasar pengembangan aplikasi game sejenis yang lebih menarik.

Kata kunci: First person shooter, non playable character, musuh, finite state machine, realitas virtual, Survival Maze, android.

SURVIVAL MAZE A VIRTUAL REALITY OF FIRST PERSON SHOOTER (FPS) GAME FOR ANDROID DEVICE

Student Name : Guruh Arya Senna
NRP : 5113100010
Major : Informatika FTIK-ITS
Advisor I : Dr.Eng. Darlis Herumurti, S.Kom.,
M.Kom.
Advisor II : Imam Kuswardayan, S.Kom., M.T.

Abstract

Game is one medium of entertainment that is rapidly growing and playing games is one way to fill the spare time. Until now a lot of types of games that already exist, in addition to the increasing features are also builders and innovation support for a game. Games that until now developed among first person shooter genre games (FPS).

The game builder features are always enhanced to attract more attention and make players / gamers more comfortable in playing, including developing features Non Playable Character (NPC). NPC has an essential function in the artificial world of the game so it become more immersive, consists of various kinds, one of which greatly adds to the challenge when playing the enemy characters. In addition, currently developing new innovations support a game, especially for games on computers and mobile devices, one of which is a technology called virtual reality.

'Survival Maze' is an FPS innovation designed based on the Virtual Reality of Android Devices. As with other virtual reality applications, this game will use the first person's perspective, plus there are enemy characters to make the game more interesting and challenging for players. The enemy

character will be equipped with Finite State Machine (FSM) method in making its behavior.

Testing the application is done by operating on different devices and also through questionnaires of respondents who have played a similar game, the type of FPS or type of maze / labyrinth before. Based on testing, it is shown that this application can show the objects in virtual reality view and there is no functional requirements that is not working as the base features in this game. With the development of this application, it is expected to be an alternative game and become the basis of development of similar game applications to become more interesting.

Key words: *First person shooter, non playable character, enemy, finite state machine, virtual reality, Survival Maze, android.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur bagi Allah Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **Permainan *First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze***. Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari. Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah Tuhan Yang Maha Esa atas anugerahnya yang tidak terkira kepada penulis.
2. Kedua orang tua penulis, Bapak Rudi Hermawan dan Ibu Siti Khotimah yang selalu mencurahkan doa, dukungan semangat, perhatian, serta kasih sayang pada penulis.
3. Bapak Dr.Eng. Darlis Herumurti, S.Kom, M.Kom. selaku pembimbing Tugas Akhir pertama yang telah memberikan arahan dalam mengerjakan Tugas Akhir ini.
4. Bapak Imam Kuswardayan, S.Kom., M.T. selaku pembimbing Tugas Akhir kedua yang dengan sabar membimbing penulis dalam mengerjakan Tugas Akhir ini.
5. Teman-teman NDANG KAWIN!, semoga segera tercapai.
6. Semua yang dari awal hingga akhir sekarang menjadi Barocca, yang selalu menjadi cerita tersendiri di hari nanti.
7. Seluruh rekan-rekan #IGSahabat dan #IGSahabatJilid-2
8. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Januari 2018

Guruh Arya Senna

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrakvii	
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	4
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Unity 3D.....	7
2.2 Bahasa Pemrograman C#	7
2.3 Android SDK.....	8
2.4 <i>Maze Game</i>	8
2.5 <i>First Person Shooter</i>	9
2.6 <i>Finite State Machine</i>	10
2.7 Realitas Virtual.....	10
BAB III ANALISIS DAN PERANCANGAN.....	13
3.1 Analisis Perangkat Lunak.....	13
3.1.1 Deskripsi Umum Perangkat Lunak.....	14
3.1.2 Spesifikasi Kebutuhan Perangkat Lunak	16
3.1.3 Karakteristik Pengguna.....	17
3.2 Perancangan Perangkat Lunak.....	18
3.2.1 Perancangan Proses	18
3.2.2 Perancangan Skenario Aturan Main	19
3.2.3 Perancangan Diagram Kasus Penggunaan.....	22

3.2.4	Spesifikasi Skenario Kasus Penggunaan.....	22
3.2.5	Perancangan Antarmuka Pengguna	26
3.2.6	Proses Perancangan Data	29
BAB IV IMPLEMENTASI.....		45
4.1	Lingkungan Implementasi	45
4.2	Implementasi Pembuatan Aplikasi	46
4.2.1	Implementasi Antarmuka Menu.....	46
4.2.2	Implementasi Realitas Virtual dengan Cardboard ...	49
4.2.3	Implementasi Pembuatan <i>Scene</i>	50
4.2.4	Implementasi Import Assets	50
4.2.5	Implementasi <i>Load</i> Objek ke <i>Scene</i>	50
4.2.6	Implementasi Penambahan Audio	51
4.2.7	Implementasi Pembuatan <i>Script</i>	52
4.2.8	Implementasi Kontrol Aplikasi.....	53
4.2.9	Implementasi Mendapatkan Informasi.....	58
4.3	Implementasi Aplikasi Permainan	61
4.3.1	Implementasi Finite State Machine Karakter NPC..	61
4.3.2	Implementasi Menelusuri Labirin dan Mengambil Barang Bantuan	66
4.3.3	Implementasi First Person Shooter	67
4.3.4	Implementasi Menemukan Finish.....	71
BAB V PENGUJIAN DAN EVALUASI		73
5.1	Lingkungan Uji Coba	73
5.2	Skenario dan Hasil Uji Coba	74
5.2.1	Pengujian Fungsionalitas	74
5.2.2	Pengujian Pengguna.....	78
5.3	Evaluasi Pengujian.....	80
5.3.1	Evaluasi Pengujian Fungsionalitas.....	80
5.3.2	Evaluasi Pengujian Pengguna	81
BAB VI KESIMPULAN DAN SARAN		83
6.1.	Kesimpulan	83
6.2.	Saran	83
DAFTAR PUSTAKA.....		85
LAMPIRAN A SCREENSHOT APLIKASI		87
LAMPIRAN B KUESIONER UJI APLIKASI		95

BIODATA PENULIS.....	107
----------------------	-----

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 <i>Screenshot</i> Maze Game 3	8
Gambar 2.2 Pac-Man Google.....	9
Gambar 2.3 <i>Screenshot</i> Virtua Cop.....	9
Gambar 2.4 Contoh finite state machine	10
Gambar 2.5 Contoh penggunaan realitas virtual untuk video	11
Gambar 3.1 Diagram alur proses permainan	19
Gambar 3.2 Diagram kasus penggunaan	22
Gambar 3.3 Rancangan antarmuka halaman awal	26
Gambar 3.4 Rancangan antarmuka pilihan level.....	27
Gambar 3.5 Rancangan antarmuka tutorial	27
Gambar 3.6 Rancangan dasar antarmuka dalam permainan.....	28
Gambar 3.7 Rancangan dasar antarmuka pemain melihat karakter musuh	28
Gambar 3.8 Rancangan FSM perilaku NPC.....	30
Gambar 3.9 Rancangan FSM perilaku <i>Player</i>	31
Gambar 3.10 Ilustrasi perancangan waktu untuk interaksi objek	32
Gambar 3.11 Menampilkan Informasi Nyawa Pemain	32
Gambar 3.12 Menampilkan Informasi Atribut Bertahan Hidup Pemain	33
Gambar 3.13 Menampilkan Informasi Nyawa NPC	34
Gambar 3.14 Pengambilan Barang Bantuan untuk Pemain	35
Gambar 3.15 Menampilkan Informasi Permainan Selesai	35
Gambar 3.16 Model 2D untuk 3D labirin level 1, kiri untuk lantai 2 dan kanan untuk lantai 1	36
Gambar 3.17 Model 2D untuk 3D labirin level 2.....	37
Gambar 3.18 Model 2D untuk 3D labirin level 3.....	37
Gambar 3.19 Model 3D labirin level 1, kiri adalah lantai 2 dan kanan adalah lantai 1	38
Gambar 3.20 Model 3D labirin level 2.....	38
Gambar 3.21 Model 3D labirin level 3.....	39
Gambar 3.22 Objek NPC perempuan.....	39
Gambar 3.23 Objek NPC laki-laki	40
Gambar 3.24 Objek-objek pendukung level 1.....	40

Gambar 3.25 Objek-objek pendukung level 2	41
Gambar 3.26 Objek-objek pendukung level 3	41
Gambar 3.27 Objek-objek pendukung tampilan pemain	41
Gambar 3.28 Hasil desain level 1, lantai 2 (atas) dan lantai 1 (bawah)	42
Gambar 3.29 Hasil perancangan desain level 2	43
Gambar 3.30 Hasil perancangan desain level 3	43
Gambar 4.1 Antarmuka Halaman Menu	46
Gambar 4.2 Antarmuka tutorial	48
Gambar 4.3 Antarmuka pemilihan level	49
Gambar 4.4 Import Unity Package	49
Gambar 4.5 Import Asset ke Proyek Unity	50
Gambar 4.6 <i>Generate Collider</i> pada Model 3D	51
Gambar 4.7 Inspector Komponen Audio Source	52
Gambar 4.8 Pembuatan <i>Script</i> Baru	52
Gambar 4.9 Pengaturan Komponen pada Tombol	57
Gambar 4.10 Membuat Kotak atau Barang Bantuan	58
Gambar 4.11 Membuat Area Informasi	60
Gambar 4.12 Mengatur posisi teks informasi terhadap kamera ..	60
Gambar 4.13 Mengatur posisi teks informasi terhadap objek bergerak	60
Gambar 4.14 Pengaturan <i>Animator Controller</i>	62
Gambar 4.15 Navigation Area untuk Level 1	63
Gambar 4.16 Tampilan menu saat game <i>finish</i>	72
Gambar 4.17 Tampilan menu saat game <i>finish</i>	72
Gambar 8.1 Tampilan awal saat membuka aplikasi	87
Gambar 8.2 Tampilan menu utama	87
Gambar 8.3 Tampilan ketika <i>timer</i> berjalan untuk sebuah interaksi	88
Gambar 8.4 Tampilan halaman tutorial	88
Gambar 8.5 Tampilan halaman pilih level	89
Gambar 8.6 Tampilan awal level 1	89
Gambar 8.7 Tampilan ketika bertemu musuh dan terjadi interaksi	90
Gambar 8.8 Tampilan ketika pemain mati/ <i>game over</i>	90

Gambar 8.9 Tampilan menu ketika pemain melihat arah atas	91
Gambar 8.10 Tampilan level 1 selesai/ <i>finish</i>	91
Gambar 8.11 Tampilan awal level 2.....	92
Gambar 8.12 Tampilan level 2 selesai/ <i>finish</i>	92
Gambar 8.13 Tampilan awal level 3.....	93
Gambar 8.14 Tampilan level 3 selesai/ <i>finish</i>	93
Gambar 8.15 Tampilan saat memilih level lanjut setelah selesai level 3 (halaman tamat)	94
Gambar 9.1 Kuesioner Pengujian Aplikasi : Irfan Hanif	95
Gambar 9.2 Kuesioner Pengujian Aplikasi : Adam Widi B.....	96
Gambar 9.3 Kuesioner Pengujian Aplikasi : Arij Nafi'atul Mashuda	97
Gambar 9.4 Kuesioner Pengujian Aplikasi : Sita Nurjayanti	98
Gambar 9.5 Kuesioner Pengujian Aplikasi : Panji Rimawan.....	99
Gambar 9.6 Kuesioner Pengujian Aplikasi : Muhammad Auliaramadani	100
Gambar 9.7 Kuesioner Pengujian Aplikasi : Safira Nur Afifah	101
Gambar 9.8 Kuesioner Pengujian Aplikasi : Renita Elizabeth Sianipar.....	102
Gambar 9.9 Kuesioner Pengujian Aplikasi : Puteri Nofa A.....	103
Gambar 9.10 Kuesioner Pengujian Aplikasi : Ade Nobil M.....	104
Gambar 9.11 Kuesioner Pengujian Aplikasi : Muhammad Rofiqi	105

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna	18
Tabel 3.2 Skenario Permainan.....	21
Tabel 3.3 Spesifikasi Kasus Penggunaan UC-001	22
Tabel 3.4 Spesifikasi Kasus Penggunaan UC-002	23
Tabel 3.5 Spesifikasi Kasus Penggunaan UC-003	24
Tabel 3.6 Spesifikasi Kasus Penggunaan UC-004	25
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	45
Tabel 5.1 Lingkungan Perangkat Keras	73
Tabel 5.2 Skenario Uji Coba Fungsionalitas	74
Tabel 5.3 Hasil Uji Coba Memilih Level	75
Tabel 5.4 Hasil Uji Coba Menelusuri Tempat.....	75
Tabel 5.5 Hasil Uji Coba Melihat Informasi	77
Tabel 5.6 Hasil Uji Coba Menemukan Finish	78
Tabel 5.7 Daftar Nama Penguji Mahasiswa Teknik Informatika	79
Tabel 5.8 Daftar Nama Penguji Lainnya.....	80
Tabel 5.9 Rekapitulasi Hasil Uji Fungsionalitas	80
Tabel 5.10 Rangkuman Hasil Formulir Pengujian Aplikasi.....	81

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kontrol aksi tombol.....	47
Kode Sumber 4.2 Fungsi kembali ke menu utama.....	47
Kode Sumber 4.3 Fungsi memilih level	48
Kode Sumber 4.4 <i>Auto Click Invoke</i>	56
Kode Sumber 4.5 Jalan Otomatis	57
Kode Sumber 4.6 Ketika menabrak Barang Bantuan.....	59
Kode Sumber 4.7 Kode Program untuk NPC.....	66
Kode Sumber 4.8 Implementasi Mengambil Nilai Barang Bantuan	67
Kode Sumber 4.9 Implementasi Kode Program <i>Player</i>	69
Kode Sumber 4.10 Implementasi Kode Program <i>Player</i>	71

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Game merupakan salah satu media hiburan yang sedang pesat-pesatnya berkembang untuk saat ini, dan bermain *game* merupakan salah satu cara untuk mengisi waktu luang. Hingga saat ini banyak sekali jenis *game* yang sudah ada, selain itu semakin bertambah pula fitur pembangun maupun inovasi pendukung untuk suatu *game*. *Game* yang hingga kini berkembang diantaranya *game* bergenre *first person shooter*. [1]

Fitur-fitur pembangun *game* selalu ditingkatkan agar semakin menarik perhatian dan membuat pemain/*gamer* lebih betah dalam bermain, diantaranya berkembang fitur *Non Playable Character* (NPC). [2] NPC memiliki fungsi esensial dalam artifisial dunia *game* sehingga lebih *immersiv*e, terdiri beragam macam, salah satunya yang sangat menambah tantangan saat bermain yaitu karakter musuh. Selain itu, saat ini sedang berkembang inovasi-inovasi baru pendukung sebuah *game*, terutama untuk *game* pada komputer maupun perangkat bergerak, salah satunya adalah teknologi bernama *virtual reality*/realitas virtual. [3]

‘*Survival Maze*’, terdiri dari kata ‘*survival*’ yang artinya bertahan untuk hidup dan kata ‘*maze*’ yang artinya labirin, merupakan inovasi FPS yang akan dirancang berbasis Realitas Virtual Perangkat Andoid. Sebagaimana aplikasi realitas virtual lainnya, *game* ini akan menggunakan sudut pandang orang pertama, ditambah terdapat karakter musuh agar *game* lebih menarik dan menantang bagi pemain. Karakter musuh nantinya akan dilengkapi dengan metode *Finite State Machine* (FSM) dalam pembuatan perilakunya. [4]

Inovasi yang akan menjadi fokus dari pengajuan Tugas Akhir ini adalah *gameplay* baru pada *maze puzzle game* yang biasanya 2D menjadi FPS 3D menggunakan realitas virtual untuk

perangkat Android dan implementasi perilaku karakter musuh menggunakan FSM. Pemain akan melakukan eksplorasi peta labirin dengan bertahan hidup dari musuh yang akan menyerang apabila pemain semakin dekat dengan musuh. Dengan inovasi *gameplay* ini, diharapkan akan menjadi suatu pengalaman yang baru, membuat pemain tidak mudah merasa bosan atau jenuh dalam memainkan *maze game* tersebut. Dengan adanya aplikasi ini, diharapkan memberi dampak positif terhadap perkembangan *game*, sehingga muncul berbagai inovasi-inovasi lainnya dalam bidang *game* sehingga lebih memberikan banyak pilihan bagi pemain.

1.2 Rumusan Masalah

Berikut rumusan masalah yang diangkat dalam tugas akhir ini adalah :

1. Bagaimana merancang *gameplay maze game* dengan fitur 3D *First Person Shooter* (FPS) menggunakan konsep Realitas Virtual Android?
2. Bagaimana merancang mekanisme perilaku karakter musuh dengan metode *Finite State Machine* (FSM)?
3. Bagaimana implementasi dari rancangan di atas diselesaikan dengan *Game Engine Unity*?

1.3 Batasan Masalah

Berikut batasan masalah pada tugas akhir ini adalah:

1. Aplikasi yang dibuat merupakan aplikasi yang berjalan di Android dengan sensor *gyroscope*.
2. Aplikasi yang dibuat merupakan aplikasi realitas virtual yang membutuhkan perangkat kepala untuk realitas virtual tanpa membutuhkan alat kontrol (*input*) lain.
3. Lingkungan pengembangan yang digunakan menggunakan aplikasi *Unity 3D Free License* dan bahasa pemrograman C#.
4. Metode dalam implementasi karakter musuh menggunakan *Finite State Machine*.

5. Kondisi untuk musuh terbatas hanya untuk diam, bergerak mendekat/mengejar pemain, dan menyerang pemain.
6. Obyek musuh dibagi menjadi dua jenis, yaitu musuh yang bergerak cepat namun lemah dan musuh yang bergerak lambat namun lebih kuat.
7. Pemain hanya memiliki 3 nyawa, yang apabila berhasil diserang NPC maka langsung kehilangan 1 nyawa dari total yang ada sebelumnya dan permainan akan berakhir ketika pemain tidak memiliki sisa nyawa (nilai nyawa = 0).
8. Pemain memiliki pertahanan untuk melawan NPC dengan senjata yang otomatis menembak dalam waktu tertentu dan memiliki waktu pulih (*cooling down*) tertentu pula.
9. Pemain memiliki kesempatan untuk mendapat barang/*item* amunisi untuk menambah persediaan amunisi senjata yang dapat ditemui pada jalan yang dilewati pemain.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membuat aplikasi realitas virtual Android yang mengimplementasikan inovasi baru dalam *gameplay maze puzzle game* yang menerapkan mekanisme *Finite State Machine* untuk karakter musuh yang ada di dalamnya.

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah :

1. Memberikan suasana *gameplay* baru dengan menambah fitur *First Person Shooter* dan mengimplementasikan metode *Finite State Machine* untuk karakter musuh pada *maze virtual reality game* Android.
2. Mengasah daya ingat pemain dan kemampuan dalam menyusun strategi untuk menyelesaikan/memenangkan permainan.
3. Sebagai sarana hiburan untuk para pengguna.

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

- A. Penyusunan proposal tugas akhir
Proposal tugas akhir berisi deskripsi pendahuluan tugas akhir yang akan dikerjakan. Bagian pendahuluan terdiri dari latar belakang tugas akhir, rumusan masalah, batasan masalah, tujuan pengerjaan tugas akhir, dan manfaat dari hasil pengerjaan tugas akhir. Selain itu, dalam proposal tugas akhir juga dijabarkan tinjauan pustaka yang digunakan sebagai referensi pendukung pengerjaan tugas akhir. Dalam proposal ini juga terdapat penjelasan mengenai metodologi yang dipakai, mulai dari tahap penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat juga sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.
- B. Studi literatur
Tahap studi literatur merupakan tahap pencarian informasi, pembelajaran dan studi literatur sejumlah referensi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan informasi dan pelatihan-pelatihan untuk pemahaman dasar teori yang digunakan pada topik tugas akhir, seperti pemodelan tiga dimensi (3D), realitas virtual, Google Cardboard, *maze game*, *Finite State Machine*, Unity, dan aplikasi perangkat bergerak khususnya Android.
- C. Perancangan perangkat lunak
Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik tugas akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang akan digunakan dan proses-proses yang akan dilaksanakan, yang meliputi perancangan objek 3D dari objek labirin 2D area permainan dan perancangan skema level untuk membuat permainan semakin menarik.

- D. Implementasi perangkat lunak
Aplikasi ini dibangun menggunakan Game Engine Unity 3D Free, dengan bahasa pemrograman C#. sedangkan untuk pembuatan *environment* dibuat menggunakan aplikasi Blender dan juga dari Asset Store Unity.
- E. Uji coba dan evaluasi
Pada tahap ini dilakukan uji coba dengan indikator untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional selain itu juga dilakukan pengujian non-fungsionalitas apakah aplikasi sudah bisa memberi kenyamanan dan kesan *immersive* kepada pengguna. Pengujian aplikasi ini dilakukan dengan metode kuesioner kepada pengguna yang sebelumnya telah diminta untuk menjalankan aplikasi secara langsung.
- F. Penyusunan laporan tugas akhir
Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

1.7 Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan objek, implementasi pembuatan aplikasi, dan implementasi pembuatan simulasi.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Unity 3D

Unity adalah ekosistem pengembangan game yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. Unity merupakan *easy multiplatform publishing*. Unity memiliki *Asset* yang menyediakan *Asset* untuk di unduk secara gratis maupun berbayar. Terdapat pula *Unity Community* yang menyediakan tutorial secara gratis untuk semua pengguna unity. Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, yaitu JavaScript, C#, dan Boo. *Flexible and EasyMoving*, *rotating*, dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating*, *removing*, dan *changing properties*. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET. Artinya perjalanan program dilakukan dengan *Open Source .NET platform*, Mono. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Ninetendo, Oculus VR dan Intel.

2.2 Bahasa Pemrograman C#

C# (dibaca: c *sharp*) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .Net *framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic dan lain-lain dengan beberapa penyederhanaan.

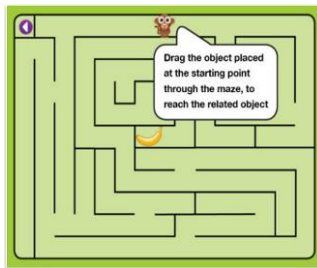
2.3 Android SDK

Android-SDK merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Android Studio. Dengan menggunakan android studio ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi Android, dll.

2.4 Maze Game

Maze game adalah *video game* bergenre diskripsi pertama yang digunakan oleh wartawan selama tahun 1980 untuk menggambarkan permainan apapun di mana seluruh lapangan adalah labirin. Keputusan cepat pemain diperlukan untuk melarikan diri dari monster, berlomba dengan lawan, atau menavigasi labirin dalam batas waktu.

Salah satu contoh *maze game* adalah Maze Game 3. Maze Game 3 adalah salah satu permainan edukasi yang tersedia di App Store untuk perangkat iOS yang dibuat oleh MyFirstApp Ltd. Tampilan bermain pada Maze Game 3 dapat dilihat pada Gambar 2.1.



Gambar 2.1 Screenshot Maze Game 3

Contoh lain dari *maze game* yang populer adalah permainan Pac-Man. Pac-Man adalah permainan yang sudah ada sejak 1980 dan dibuat oleh Namco. Berikut Gambar 2.2 adalah permainan Pac-Man yang akan muncul ketika memberikan kata kunci pencarian “pacman” dengan mesin pencarian *google.com*.



Gambar 2.2 Pac-Man Google

2.5 *First Person Shooter*

Pengembangan *First Person Shooter* (FPS) dimulai pada tahun 1973, dan 1974, Spasim. Setelah itu, lebih banyak judul bermunculan seperti MIDI Maze tahun 1987, kemudian bersatu dengan genre kekerasan membentuk Wolfenstein 3D 1992, akan tetapi lebih kalah populer dengan *game* Doom. Half Life (1998) dan Half Life 2 (2004) meningkatkan narasi dan elemen teka-teki (puzzle).

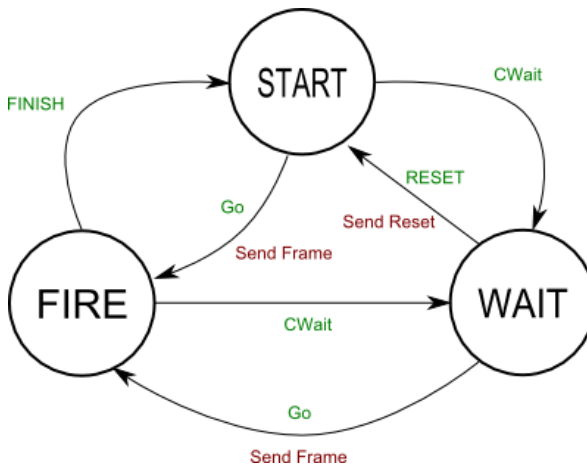
Contoh lain dari permainan FPS yang pernah populer pada masanya adalah Virtua Cop. Virtua Cop dibuat oleh Sega AM2 yang dikepalai Yu Suzuki. Tampilan bermain Virtua Cop dapat dilihat pada Gambar 2.3.



Gambar 2.3 Screenshot Virtua Cop

2.6 *Finite State Machine*

Finite State Machine (FSM) adalah model dari sistem komputasi, terdiri dari satu set keadaan (termasuk keadaan mulai), alfabet dari simbol-simbol yang berfungsi sebagai seperangkat kemungkinan masukan ke mesin, dan fungsi transisi yang memetakan setiap keadaan ke keadaan lain (atau keadaan itu sendiri) untuk setiap simbol masukan yang diberikan. Mesin beroperasi dengan diberi serangkaian simbol, dan bergerak mealui serangkaian keadaan. Studi tentang kekuatan komputasi dari FSM dan mesin terkait lainnya sangat penting dalam ilmu komputer dan linguistik.



Gambar 2.4 Contoh finite state machine

2.7 *Realitas Virtual*

Realitas virtual adalah simulasi realitas dan mendalam dari lingkungan tiga dimensi, dibuat dengan menggunakan perangkat lunak interaktif dan perangkat keras, dan dapat dikendalikan oleh gerakan tubuh. Realitas virtual dapat diterapkan dalam berbagai hal dan aspek, diantaranya pendidikan dan pelatihan, arsitektur

dan desain perkotaan, *video games* dan film, arkeologi, serta medis.

Berikut Gambar 2.5 adalah contoh penerapan realitas virtual untuk video yang ditampilkan dalam artikel 6 November 2015, ditulis oleh Jacob Kleinman di technobuffalo.com



Gambar 2.5 Contoh penggunaan realitas virtual untuk video

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis dan perancangan aplikasi permainan realitas virtual ‘Survival Maze’ dengan mengimplementasikan *finite state machine* pada karakter musuh. Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Perangkat Lunak

Sudah banyak kita jumpai *game* berbasis labirin atau *maze*, seperti salah satunya yang terkenal ialah Pacman. Namun, kebanyakan dari *game* tersebut mempunyai desain yang sederhana dan model-model yang masih 2D. Sehingga pengguna bisa cepat merasa bosan ketika sudah lama bermain. Memang sudah ada beberapa *game* yang telah menerapkan model 3D untuk permainannya.

Di sisi lain terdapat jenis permainan *first person shooter*, misalnya Virtua Cop yang dulu pernah populer untuk permainan di komputer. Permainan jenis FPS yang menggunakan sudut pandang orang pertama juga sudah banyak dibuat dan dikembangkan untuk para penggila *game* terutama karena meningkatkan *adrenaline* ketika pemain berinteraksi baik dengan NPC atau dengan pemain lain ketika online.

Namun diantara keduanya, masih belum memberikan pengalaman yang berbeda dalam memainkan *game* tersebut terlebih di zaman kemajuan teknologi yang semakin pesat ini. Oleh karena itu penulis mencoba menggabungkan permainan jenis *maze* dengan permainan jenis *first person shooter* yang dipadukan lagi dengan teknologi *virtual reality* untuk memberikan pengalaman bermain yang lebih menarik, tidak membosankan dan terasa lebih nyata karena menerapkan realitas virtual.

Aplikasi ini dibangun dengan teknologi Google Cardboard sehingga pengguna dapat menjalankannya pada perangkat Android yang didukung dengan sensor *gyroscope*. Aplikasi ini akan dibuat dengan menggunakan aplikasi Blender untuk pembuatan objek 3D dan Unity karena kedua aplikasi ini telah menyediakan fitur yang mendukung Google Cardboard untuk realitas virtual. Aplikasi ini menerapkan realitas virtual, sehingga diharapkan pengguna dapat melihat dan merasakan pengalaman bermain yang nyata, tentunya ditambah dengan fitur seperti pada permainan FPS yaitu adanya NPC (dalam hal ini musuh) sehingga dalam menelusuri labirin tersebut terasa lebih menantang.

3.1.1 Deskripsi Umum Perangkat Lunak

Survival Maze merupakan *maze game* yang ditambah fitur *game First Person Shooter* (FPS) menggunakan inovasi Realitas Virtual untuk Android dengan mekanisme *Finite State Machine* pada karakter musuh. Fokus dalam pengerjaan Tugas Akhir ini terletak pada *gameplay*, jadi bagaimana caranya menambahkan konsep FPS dengan Realitas Virtual Android dan metode FSM pada karakter musuh untuk melengkapi *maze game*, sehingga menjadi suatu pengalaman yang baru untuk pemain, membuat pemain tidak mudah merasa bosan dan jenuh dalam memainkan *maze game* tersebut.

Untuk bermain Survival Maze nantinya, pengguna/pemain memerlukan sebuah perangkat Android dengan sensor *gyroscope* dan sebuah perangkat kepala (*headset*) realitas virtual yang sesuai untuk perangkat Android tersebut. Ketika selesai menyiapkan alat yang diperlukan, pemain dapat menggunakan/bermain dengan lebih nyaman.

Saat memasuki menu, pemain harus menahan arah sasaran beberapa saat pada menu yang diinginkan untuk memilih menu tersebut. Dalam hal ini contohnya adalah memilih menu 'Play'.

Game Survival Maze ini bercerita tentang seseorang (*player*) yang terjebak pada suatu daerah seperti labirin dan harus mencari jalan keluar dengan menyusuri jalan yang dia pilih serta menghindari musuh yang mencoba membunuhnya. Pemain dapat berjalan dan melihat sekitar dengan mengarahkan pandangan/orientasi dirinya sesuai keinginannya. Pemain memiliki Nyawa sebanyak 3 poin. Waktu akan berjalan ketika pemain mulai bergerak dari titik awalnya.

Setiap karakter bertemu dengan musuh dan sasaran pemain diarahkan kepada musuh, maka secara otomatis tangan pemain akan terlihat membawa senjata dan bersiap menyerang/menembak musuh. Sasaran yang tadinya berwarna merah akan menjadi berwarna hijau ketika ditahan beberapa saat ke arah musuh dan secara otomatis akan menyerang/menembak musuh. Setiap selesai satu kali serangan, sasaran akan berwarna merah kembali dan pemain harus menunggu beberapa saat lagi untuk dapat melakukan serangan selanjutnya. Senjata pemain pada awalnya memiliki amunisi yang terbatas, namun di dalam perjalanannya, pemain akan menemukan amunisi tambahan untuk senjatanya. Karakter musuh yang akan dibuat dalam permainan ini dibagi menjadi 2 jenis, yaitu karakter musuh yang cepat namun lemah dan karakter musuh yang lambat namun lebih kuat. Posisi awal musuh akan diam di tempat ketika berada pada jarak yang cukup jauh dari pemain, bergerak dan memburu pemain apabila pemain sudah memasuki jarak pandang musuh.

Ketika pemain dikejar musuh dan tertangkap, *Health Point* (HP) pemain akan terus berkurang sampai habis dalam beberapa waktu yang telah ditentukan sebagai skenario musuh yang sedang menyerang dan kehilangan 1 dari sisa Nyawa yang dimilikinya jika HP telah habis. Jika jumlah Nyawa pemain sudah 0, maka pemain dinyatakan kalah dan permainan dimulai kembali dari awal.

3.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Kebutuhan sistem yang dibuat melibatkan dua hal, yaitu kebutuhan fungsional dan non-fungsional. Dua kebutuhan tersebut akan dijelaskan lebih lanjut pada Subbab 3.1.2.1 dan Subbab 3.1.2.2.

3.1.2.1 Spesifikasi Kebutuhan Fungsional

Pada aplikasi ini ada beberapa kebutuhan fungsionalitas yang mendukung untuk berjalannya aplikasi. Kebutuhan fungsionalitas pada aplikasi ini antara lain sebagai berikut :

1. Memilih tingkat kesulitan bertahan hidup

Aplikasi memiliki fungsionalitas dapat memilih tingkatan kesulitan untuk dicoba. Pengguna harus menyelesaikan setiap level mulai dari level 1 hingga selesai. Tingkatan pada aplikasi ini sendiri terbagi menjadi tiga. Terdapat tingkat kesulitan dari tingkatan maze mudah, sedang dan paling sulit. Setelah melewati level yang mudah maka akan dilanjutkan dengan tingkatan yang lebih sulit.

2. Pemain dapat melakukan interaksi dengan otomatis (dengan *timer*) sesuai dengan jenis objek yang dipilih

Aplikasi dapat menampilkan interaksi yang tepat untuk beberapa jenis objek dalam aplikasi yang dipilih oleh pemain. Interaksi yang dimaksud adalah interaksi antara pemain dengan NPC serta interaksi antara pemain dengan tombol-tombol dalam aplikasi.

3. NPC dapat bertindak sesuai *state*/kondisi yang sedang terjadi sesuai skema pembuatan

Aplikasi dapat menampilkan beberapa kondisi musuh dengan batasan-batasan yang sudah ditentukan, dengan memberikan kondisi-kondisi tertentu yang akan membuat trigger pada NPC. Hal ini adalah pengaplikasian dari finite state machine sehingga dapat membuat karakter NPC menjadi karakter kecerdasan buatan sederhana.

4. Pemain dapat mengeksplor arena labirin dan berusaha bertahan hidup

Aplikasi dapat menampilkan *scene* dimana pemain mencoba bertahan hidup dengan apa yang dimiliki saat itu dan apa yang ditemukan di jalan. Pengguna memiliki atribut untuk memperlihatkan statusnya dan melakukan eksplorasi pada tiap tingkatan level, mulai dari tingkatan paling mudah hingga paling sulit.

3.1.2.2 Spesifikasi Kebutuhan Non-Fungsional

Terdapat beberapa kebutuhan non-fungsional. Apabila kebutuhan non-fungsionalitas dipenuhi, dapat meningkatkan kualitas dari aplikasi ini. Berikut daftar kebutuhan non-fungsional:

1. Kebutuhan grafis

Grafis yang bagus dapat menambahkan nilai yang lebih terhadap suatu aplikasi. Kemiripan suatu objek pada aplikasi ini dengan objek nyatanya dapat membuat pengguna memiliki pengalaman yang lebih maksimal dalam permainan.

2. Aspek usability

Pada aspek ini indikator yang digunakan adalah kemudahan penggunaan menu, kemudahan dalam mendapatkan informasi, dan kesan *immersive*, yaitu membuat pemain nyaman dalam bermain.

3.1.3 Karakteristik Pengguna

Berdasarkan deskripsi umum perangkat lunak pada Subbab 3.1.1, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini hanya satu orang, yaitu pengguna yang menjalankan aplikasi. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Pengguna	Pihak luar yang mencoba aplikasi	Menjalankan aplikasi	Tidak ada

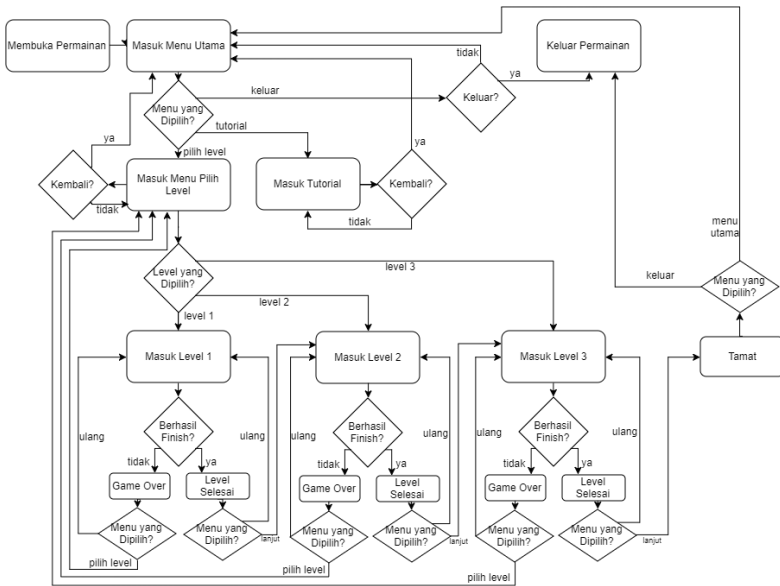
3.2 Perancangan Perangkat Lunak

Tahap perancangan pada bab ini dibagi menjadi beberapa bagian yaitu perancangan proses, perancangan alur permainan, perancangan diagram kasus penggunaan, spesifikasi skenario kasus penggunaan, perancangan antarmuka aplikasi, dan perancangan data.

3.2.1 Perancangan Proses

Pada subbab ini akan dijelaskan mengenai rancangan proses yang akan dilakukan untuk mendukung kebutuhan fungsionalitas perangkat lunak yang sudah dirancang. Proses ini penting agar perangkat lunak dapat berjalan secara baik dan lancar.

Secara umum alur proses pada *game* ini hanya satu, yaitu pengguna akan membuka permainan dan memasuki menu utama, lalu akan terjadi percabangan apakah pengguna akan memulai permainan atau membuka tutorial atau keluar permainan. Jika pengguna memilih untuk memulai permainan, maka permainan akan dimulai dan akan kembali menuju menu utama jika pemain memilih menu kembali saat bermain. Jika pengguna memilih menu tutorial, maka pengguna akan menuju halaman tutorial dan akan kembali menuju menu utama jika pemain memilih menu kembali saat berada di halaman tutorial. Jika pengguna memilih keluar saat berada di menu utama, maka permainan akan ditutup/keluar. Selengkapny, alur proses ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram alur proses permainan

3.2.2 Perancangan Skenario Aturan Main

Pada aplikasi ini menerapkan aturan main dengan cara bertahan hidup dan menemukan jalan keluar dari labirin. Dalam desain level yang akan dibuat, permainan akan dibentuk menjadi sebuah cerita dengan tujuan akhirnya adalah pemain dapat berpindah menuju tempat baru yang lebih aman, tetapi pemain harus bertahan melewati tempat-tempat yang berbahaya, yang dipenuhi musuh.

Rincian lebih lanjut, permainan akan dibuat menjadi 3 level, dengan level pertama berlatar belakang perkantoran dan dengan tingkat kesulitan paling mudah. Pada level ini, pemain berada di lantai kedua gedung dan harus mencari jalan menuju lantai pertama dan keluar dari gedung, tentunya dengan rintangan musuh-musuh yang ada di gedung. Level selanjutnya berlatar belakang perkotaan, yaitu setelah keluar dari kantor (level

pertama) dengan area permainan yang lebih luas dan tingkat kesulitan yang lebih dari level sebelumnya. Untuk level terakhir berlatarkan pedesaan, yaitu setelah keluar dari perkotaan, dengan luas area permainan yang hampir sama dengan level kedua dan dengan tantangan paling sulit diantara semua level yang ada. Tantangan dan kesulitan bergantung dari jumlah musuh dan jenis musuh yang ada di setiap levelnya.

Langkah pertama yang harus dilakukan untuk memulai permainan ini adalah memilih level antara level 1, level 2, atau level 3. Setelah memilih level, sistem akan langsung masuk ke permainan. Dalam permainan, kontrol yang digunakan adalah dengan mengarahkan kamera ke arah lantai untuk bergerak maju atau berjalan, lalu mengarahkan kamera ke objek selain lantai untuk berhenti. Pengguna harus bertahan hidup dan menemukan garis finish untuk menyelesaikan permainan ini dengan menelusuri labirin dan dengan melawan atau menghindari musuh. Cara melawan musuh adalah dengan mengarahkan kamera ke arah musuh, menunggu hingga *timer* penuh sehingga akan terjadi satu kali tembakan di setiap satu kali putaran *timer*. Jumlah peluru terbatas sehingga pengguna harus memiliki strategi tepat untuk menyelesaikan level yang dipilih. Saat menelusuri labirin pengguna diminta untuk tetap bertahan hidup sampai menemukan jalan keluar. Dalam perjalanannya menelusuri labirin, pengguna dapat menemukan barang-barang bantuan agar pengguna dapat bertahan hidup lebih lama. Barang-barang yang dimaksud adalah kotak amunisi dan kotak obat. Kotak amunisi berguna untuk menambah jumlah peluru senjata pengguna dan kotak obat berguna untuk menambah nilai dari nyawa pengguna.

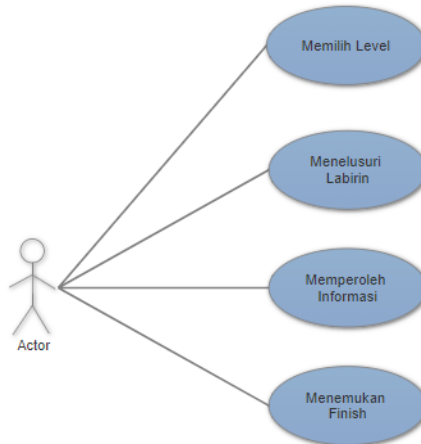
Kondisi kalah atau menang dalam permainan ini yaitu, pengguna akan menang dan menyelesaikan suatu level jika mampu bertahan hidup dan mencapai *finish*. Kemudian pengguna akan kalah dan diminta mengulang pada level tersebut jika mati karena NPC dan tidak bisa mencapai *finish*. Skenerio permainan dapat dilihat pada Tabel 3.2

Tabel 3.2 Skenario Permainan

Spesifikasi	Keterangan	
Nama Skenario	Aturan Permainan	
Deskripsi	Skeneraio ini berfungsi agar aktor dapat menyelesaikan aplikasi permainan	
Kondisi Awal	Objek sudah tampil pada aplikasi	
Kondisi Akhir	Pindah ke <i>scene level</i> selanjutnya	
Aktor	Pengguna	
Alur Utama	Aktor	Sistem
	Mengarahkan perangkat pada objek tertentu	Menampilkan objek sesuai arah sudut <i>smartphone</i>
	Mengarahkan perangkat pada sudut tertentu (kondisi berhenti)	Menggerakkan kamera ke depan
	Mengarahkan perangkat pada sudut tertentu (kondisi berjalan)	Menghentikan gerak kamera ke depan
	Mengarahkan kamera ke NPC	Menggerakkan kamera ke NPC (<i>trigger timer</i> untuk menembak otomatis)
	Mengarahkan kamera ke barang bantuan	Menggerakkan kamera ke barang batuan (mengambil nilai barang bantuan tersebut)
	Mengarahkan kamera ke objek finish	Menggerakkan kamera menuju garis finish
	Menabrak objek finish	Menampilkan informasi permainan selesai
Alur Alternatif	Aktor	Sistem
	Berhenti di titik tertentu dan melakukan <i>trigger</i> pada menu yang tersedia	Menjalankan perintah sesuai menu yang dipilih

3.2.3 Perancangan Diagram Kasus Penggunaan

Dalam aplikasi tugas akhir ini, terdapat empat kasus penggunaan yang ada yaitu memilih level, menelusuri labirin, memperoleh informasi, dan menemukan *finish*. Rancangan kasus penggunaan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram kasus penggunaan

3.2.4 Spesifikasi Skenario Kasus Penggunaan

Setelah diketahui bagaimana perancangan kasus penggunaan, maka spesifikasi kasus penggunaan akan dijelaskan pada subbab berikut.

3.2.4.1 Memilih Level

Spesifikasi kasus penggunaan Memilih Level dapat dijabarkan pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kasus Penggunaan UC-001

Spesifikasi	Keterangan
Kode Kasus Penggunaan	UC-001
Nama Kasus Penggunaan	Memilih Level

Deskripsi	Kasus pengguna ini berfungsi supaya aktor dapat memilih level yang diinginkan	
Kondisi Awal	Aplikasi sudah berjalan dan menampilkan pilihan level	
Kondisi Akhir	<i>Scene</i> level yang dipilih terbuka	
Aktor	Pengguna	
Alur Utama	Aktor	Sistem
	Mengarahkan <i>reticle</i> ke menu level yang dipilih	Muncul timer untuk memilih level yang dipilih
	Menunggu <i>timer</i> untuk <i>trigger</i> memilih level	<i>Load</i> dan <i>render</i> objek
Alur Alternatif	Aktor	Sistem
	Mengarahkan <i>reticle</i> ke tombol menu utama	Muncul timer untuk memilih tombol menu utama
	Menunggu <i>timer</i> untuk <i>trigger</i> memilih tombol menu utama	<i>Load scene</i> menu utama

3.2.4.2 Menelusuri Labirin

Spesifikasi kasus penggunaan Menelusuri Labirin dapat dijabarkan pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kasus Penggunaan UC-002

Spesifikasi	Keterangan	
Kode Kasus Penggunaan	UC-002	
Nama Kasus Penggunaan	Menelusuri Labirin	
Deskripsi	Kasus pengguna ini berfungsi supaya aktor dapat berjalan dan melihat-lihat di lingkungan dalam permainan	
Kondisi Awal	Objek sudah tampil pada aplikasi	
Kondisi Akhir	-	
Aktor	Pengguna	
Alur Utama	Aktor	Sistem

	Mengarahkan target/sasaran pada objek tertentu	Menampilkan objek sesuai arah sudut <i>smartphone</i>
	Mengarahkan perangkat pada sudut tertentu (kondisi berhenti)	Menghentikan gerak kamera ke depan
	Mengarahkan perangkat pada sudut tertentu (kondisi berjalan)	Menggerakkan kamera ke depan
Alur Alternatif	Aktor	Sistem
	Berhenti di titik tertentu sampai nyawa habis	Memulai ulang <i>scene stage</i> sekarang

3.2.4.3 Memperoleh Informasi Atribut *Player* dan NPC

Spesifikasi kasus penggunaan Memperoleh Informasi dapat dijabarkan pada Tabel 3.5.

Tabel 3.5 Spesifikasi Kasus Penggunaan UC-003

Spesifikasi	Keterangan	
Kode Kasus Penggunaan	UC-003	
Nama Kasus Penggunaan	Memperoleh informasi	
Deskripsi	Kasus pengguna ini berfungsi supaya aktor dapat mendapatkan informasi tentang atribut/kondisi aktor saat ini, seperti waktu, nyawa, dan amunisi senjata, serta atribut nyawa untuk NPC.	
Kondisi Awal	Teks informasi tentang waktu, nyawa, amunisi senjata, serta atribut nyawa untuk NPC terlihat	
Kondisi Akhir	Teks informasi tentang waktu, nyawa, amunisi senjata, serta atribut nyawa untuk NPC berubah nilai ketika terjadi <i>event</i> tertentu	
Aktor	Pengguna	
Alur Utama	Aktor	Sistem
	Mengarahkan perangkat pada sudut tertentu	Menampilkan semua atribut pemain atau NPC yang ditemui dan

		melakukan perubahan atribut sesuai kondisi
	Menabrak objek tertentu	Menampilkan semua atribut pemain atau NPC yang ditemui dan melakukan perubahan atribut sesuai kondisi
Alur Alternatif	Aktor	Sistem
	Mengarahkan <i>reticle</i> ke tombol kembali	Muncul timer untuk tombol kembali
	Menunggu <i>timer</i> untuk <i>trigger</i> memilih tombol kembali	Membuka halaman pilih level

3.2.4.4 Menemukan *Finish*

Spesifikasi kasus penggunaan Menemukan *Finish* dapat dijabarkan pada Tabel 3.6.

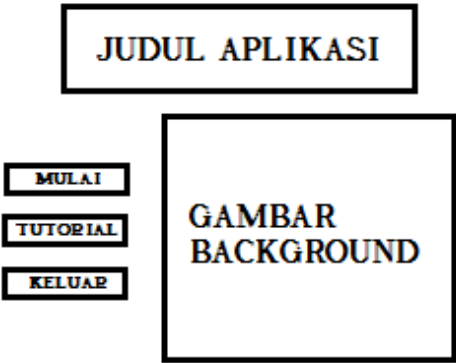
Tabel 3.6 Spesifikasi Kasus Penggunaan UC-004

Spesifikasi	Keterangan	
Kode Kasus Penggunaan	UC-004	
Nama Kasus Penggunaan	Menemukan Finish	
Deskripsi	Kasus pengguna ini berfungsi supaya aktor dapat mendapatkan informasi dan pilihan untuk melanjutkan ke level berikutnya	
Kondisi Awal	Teks informasi tidak muncul	
Kondisi Akhir	Pindah ke <i>scene level</i> selanjutnya	
Aktor	Pengguna	
Alur Utama	Aktor	Sistem
	Menabrak objek finish	Menampilkan teks informasi dan pilihan untuk ke level selanjutnya
	Mengarahkan <i>reticle</i> ke tombol level selanjutnya	Muncul timer untuk tombol level selanjutnya

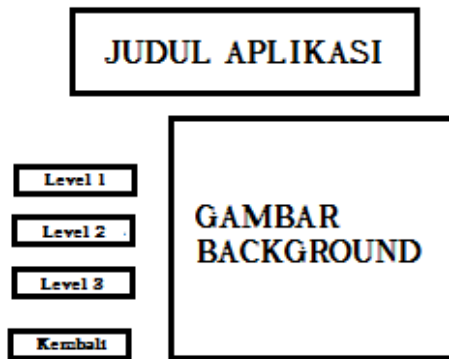
	Menunggu <i>timer</i> untuk <i>trigger</i> memilih tombol kembali	Berpindah ke <i>scene level</i> berikutnya
Alur Alternatif	Aktor	Sistem
	Mengarahkan <i>reticle</i> ke tombol <i>Restart</i>	Muncul timer untuk tombol <i>Restart</i>
	Menunggu <i>timer</i> untuk <i>trigger</i> memilih tombol <i>Restart</i>	Memulai ulang <i>scene stage</i> sekarang

3.2.5 Perancangan Antarmuka Pengguna

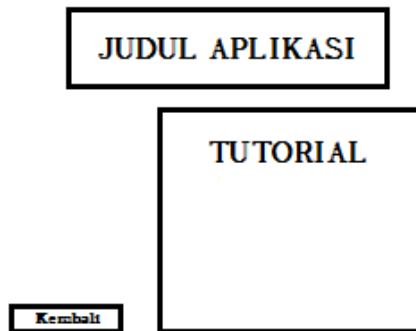
Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk Tugas Akhir. Aplikasi ini akan dibangun dengan mode normal dan mode *First Person Camera* (FPS), maka rancangan antarmuka yang dari aplikasi ini hanya berjumlah satu buah layar kosong yang akan diisi oleh objek-objek. Objek yang ditampilkan akan diisi sesuai *scene* masing-masing. Selain itu, tampilan juga ditambah dengan penampilan stereokopik agar memperdalam visual. Rancangan dari antarmuka aplikasi ini dapat dilihat pada Gambar 3.3, Gambar 3.4, Gambar 3.5, Gambar 3.6, dan Gambar 3.7.



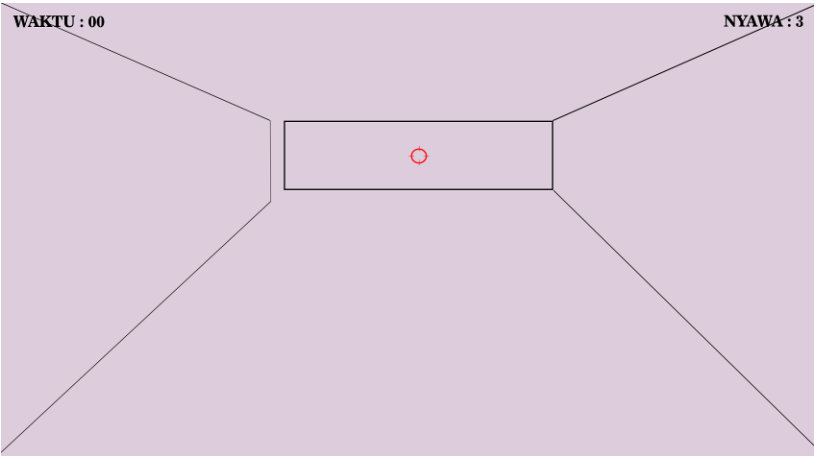
Gambar 3.3 Rancangan antarmuka halaman awal



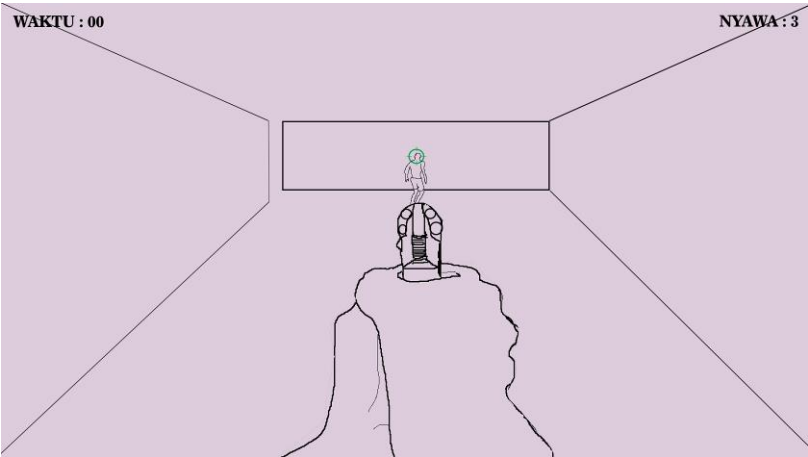
Gambar 3.4 Rancangan antarmuka pilihan level



Gambar 3.5 Rancangan antarmuka tutorial



Gambar 3.6 Rancangan dasar antarmuka dalam permainan



Gambar 3.7 Rancangan dasar antarmuka pemain melihat karakter musuh

3.2.6 Proses Perancangan Data

Perancangan data merupakan hal yang diperlukan dalam memilih dan menyusun sistem agar tepat dan dapat beroperasi secara benar. Data yang diperlukan dalam pengoperasian perangkat lunak, yaitu meliputi objek-objek yang berada pada lingkungan permainan, seperti membangun labirin yang terdiri dari dinding, lantai, serta benda-benda lain untuk membangun sebuah labirin, dan objek-objek makhluk hidup sebagai NPC (musuh) pemain. Audio juga diperlukan untuk menambah suasana yang lebih nyata pada aplikasi ini. Data mengenai lingkungan tersebut diambil dari berbagai sumber meliputi foto, video, model 3D, dan aplikasi serupa. Sedangkan data audio didapatkan dari halaman web penyedia asset audio gratis untuk game.

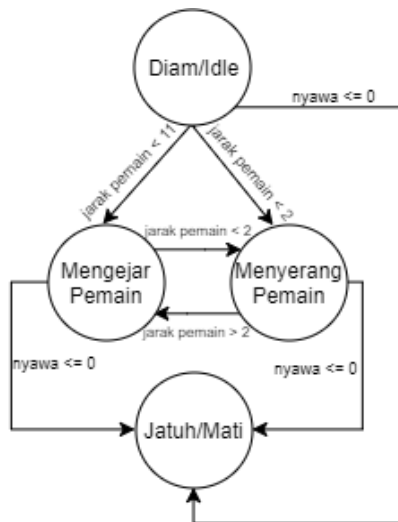
3.2.6.1 Proses Perancangan FSM

Perancangan perilaku perlu diterapkan pada *Player* dan *Non Playable Character* (NPC) sehingga nantinya dapat memberi kesan yang lebih nyata, menghibur, serta dapat memberikan tantangan dalam aplikasi permainan ini. Perilaku untuk *Player* dan NPC ini dirancang dengan metode *finite state machine* yaitu beberapa *state* (keadaan) tertentu dengan parameter-parameter tertentu pula. Proses Perancangan FSM akan dijelaskan dalam FSM perilaku NPC dan FSM perilaku *Player*.

3.2.6.1.1 FSM Perilaku NPC

Perilaku untuk NPC dibagi menjadi empat, yaitu kondisi diam, kondisi mengejar pemain dan kondisi menyerang pemain, serta kondisi mati. Kondisi utama NPC adalah diam. Ketika jarak NPC dan pemain kurang dari 11, maka NPC akan berganti kondisi menjadi mengejar pemain. Ketika jarak NPC dan pemain kurang dari 2, NPC akan berganti kondisi menjadi menyerang pemain. Jika nilai nyawa NPC mencapai 0 maka NPC akan masuk kondisi mati.

Berikut Gambar 3.8 adalah rancangan FSM untuk perilaku NPC dalam permainan dengan empat kondisi utama tersebut.

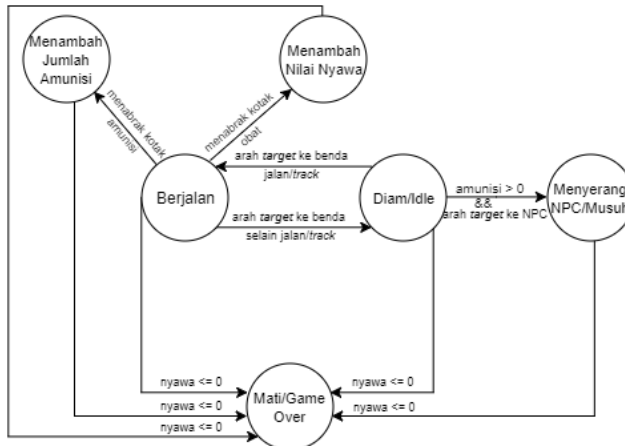


Gambar 3.8 Rancangan FSM perilaku NPC

3.2.6.1.2 FSM Perilaku *Player*

Perilaku untuk *Player* dibagi menjadi enam, yaitu kondisi diam, kondisi berjalan, kondisi kondisi menyerang NPC, kondisi menambah jumlah amunisi, kondisi menambah nilai nyawa, dan kondisi mati. Kondisi pertama *Player* ketika memulai permainan adalah diam. Ketika pandangan/target *Player* mengarah ke jalan, maka *Player* akan berganti kondisi menjadi berjalan dan berhenti kembali ketika pandangan/target tidak mengarah ke jalan. Ketika jumlah amunisi lebih dari 0 dan pandangan/target mengarah ke NPC, *Player* akan berganti kondisi menjadi menyerang NPC. Ketika *Player* menabrak kotak obat atau kotak amunisi, maka *Player* akan menambah jumlah nilai nyawa atau jumlah nilai amunisi. Jika nilai nyawa *Player* mencapai 0 maka *Player* akan masuk kondisi mati.

Berikut Gambar 3.9 adalah rancangan FSM untuk perilaku *Player* dalam permainan dengan enam kondisi utama tersebut.



Gambar 3.9 Rancangan FSM perilaku *Player*

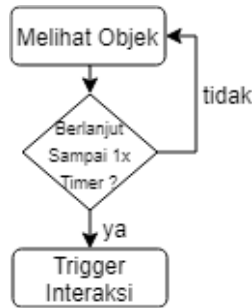
3.2.6.2 Proses Perancangan Batas Waktu (*Timer*) untuk Interaksi dengan Objek

Pembuatan batas waktu (*timer*) untuk Interaksi dengan objek adalah salah satu dari proses inti untuk permainan yang akan dibuat karena nantinya permainan akan dapat dimainkan tanpa menggunakan alat input tambahan (tanpa *controller*).

Menampilkan *timer* ini adalah untuk menentukan jenis interaksi yang akan terjadi jika sudah memasuki batas waktu yang sudah ditentukan (*trigger*). *Trigger* akan mengaktifkan fungsi-fungsi tertentu sesuai dengan objek yang diberikan perintah, misalnya menekan tombol atau melakukan interaksi dengan NPC. Tampilan *timer* akan berjalan memutar dari titik tengah target mata pemain.

Proses menampilkan batas waktu untuk interaksi dengan objek diawali dengan menentukan objek-objek apa saja yang dapat memunculkan batas waktu untuk interaksi. Selanjutnya memberikan fungsi pada setiap objek sesuai keinginan interaksi yang akan terjadi selanjutnya. Ketika bermain, cek apakah sudah terjadi satu kali putaran waktu atau tidak. Jika sudah satu kali putaran waktu, interaksi akan terjadi.

Untuk lebih jelasnya, berikut Gambar 3.10 adalah ilustrasi dari perancangan waktu untuk interaksi dengan objek.



Gambar 3.10 Ilustrasi perancangan waktu untuk interaksi objek

3.2.6.3 Proses Tampilkan Informasi Nyawa Pemain

Ketika bermain, pemain akan diberikan atribut nyawa. Pemain akan diberikan nyawa sebanyak 3 dan akan terus berkurang ketika terjadi kontak fisik dengan NPC. Jika nyawa pemain sudah mencapai angka 0 maka pemain tidak bisa melanjutkan permainan. Berikut Gambar 3.11 adalah proses menampilkan informasi nyawa pemain.



Gambar 3.11 Menampilkan Informasi Nyawa Pemain

3.2.6.4 Proses Tampilkan Informasi Atribut Bertahan Hidup Pemain

Ketika bermain, pemain akan diberikan atribut bertahan hidup. Atribut ini terdiri dari senjata dan peluru untuk senjata. Pemain akan diberikan senjata untuk bertahan hidup dengan peluru yang terbatas. Jika peluru pemain sudah mencapai angka 0 maka pemain tidak bisa menggunakan senjata di permainan. Berikut Gambar 3.12 adalah proses menampilkan informasi atribut bertahan hidup pemain.



Gambar 3.12 Menampilkan Informasi Atribut Bertahan Hidup Pemain

3.2.6.5 Proses Tampilkan Informasi Nyawa NPC (Musuh)

Ketika bermain, musuh akan diberikan atribut nyawa. Pemain dapat mengurangi nyawa NPC dengan melakukan kontak dengan senjatanya. Terdapat beberapa jenis musuh di dalam permainan, sehingga jumlah nyawa musuh akan dibuat beragam sesuai kebutuhan dan skema permainan. Jika nyawa musuh habis, musuh akan mati. Berikut Gambar 3.13 adalah proses menampilkan informasi nyawa musuh.



Gambar 3.13 Menampilkan Informasi Nyawa NPC

3.2.6.6 Proses Pengambilan Barang Bantuan untuk Pemain

Dalam permainan, pemain akan disediakan barang bantuan untuk tetap bertahan hidup. Barang bantuan ini terdiri dari peluru dan kotak obat. Apabila pemain menabrak barang bantuan ini, secara otomatis pemain akan mendapatkannya. Misalnya pemain menuju arah kotak obat, maka selanjutnya pemain akan mendapat tambahan nilai nyawa sehingga pemain dapat bertahan lebih lama ketika terjadi kontak dengan musuh. Begitu pula dengan peluru, pemain akan mendapat peluru tambahan untuk dapat digunakan melawan musuh-musuh pemain dalam permainan. Beberapa hal yang perlu diperhatikan dalam perancangan ini adalah menentukan jumlah peluru dan nilai obat untuk didapatkan pemain ketika pemain menabraknya, membuat objek untuk kotak obat dan kotak peluru, memberikan *collider* pada tiap objek. Jika pemain menabrak objek, maka nilai dari objek akan ditambahkan pada atribut pemain. Untuk lebih jelasnya, berikut Gambar 3.14 adalah proses pengambilan barang bantuan untuk pemain.



Gambar 3.14 Pengambilan Barang Bantuan untuk Pemain

3.2.6.7 Proses Tampilkan Informasi Permainan Selesai

Ketika pengguna telah menemukan jalan keluar atau garis *finish* maka sistem akan menampilkan informasi Permainan Selesai yang menunjukkan bahwa suatu level dari aplikasi permainan tersebut telah berhasil diselesaikan. Pembuatan informasi pada garis *finish* membutuhkan area untuk menampilkan informasi dan *Trigger* untuk menampilkan informasi. *Trigger* akan aktif jika kamera/pemain menabraknya, sehingga informasi Permainan Selesai akan tampil. Beberapa hal yang perlu diperhatikan dalam perancangan ini adalah membuat objek untuk garis *finish* memberi *collider* pada objek tersebut. Jika pemain menabrak objek, maka level yang sedang dimainkan akan selesai. Untuk lebih jelasnya, berikut Gambar 3.15 adalah proses menampilkan informasi Permainan Selesai.



Gambar 3.15 Menampilkan Informasi Permainan Selesai

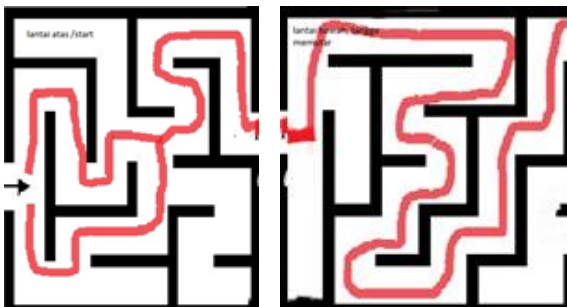
3.2.6.8 Perancangan Objek dalam *Game*

Pembuatan objek 3D dilakukan dengan memodelkan objek ke bentuk tiga dimensi terlebih dahulu. Setelah objek 3D terbentuk, selanjutnya adalah pemberian tekstur yang disesuaikan dengan objek aslinya. Sebagian besar pembuatan objek untuk skema level dibuat dari dalam Unity menggunakan alat bantu ProBuilder Basic yang dapat ditemukan dan diunduh melalui Asset Store Unity. Selain membuat menggunakan ProBuilder, kebanyakan model didapat melalui unduhan dari berbagai macam sumber.

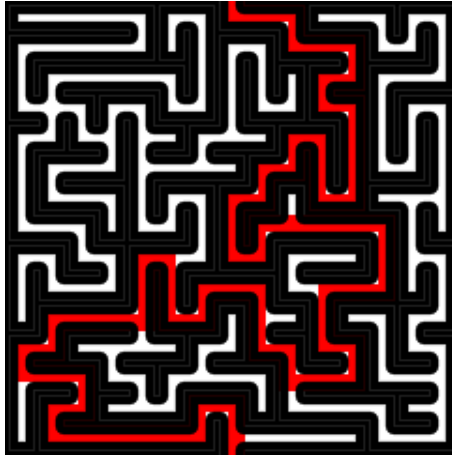
Bab ini dibagi ke dalam empat subbab meliputi Perancangan Model Labirin, Perancangan Model NPC, Perancangan Model-Model Pendukung Lain, dan Perancangan Desain Level.

3.2.6.8.1 Perancangan Model Labirin

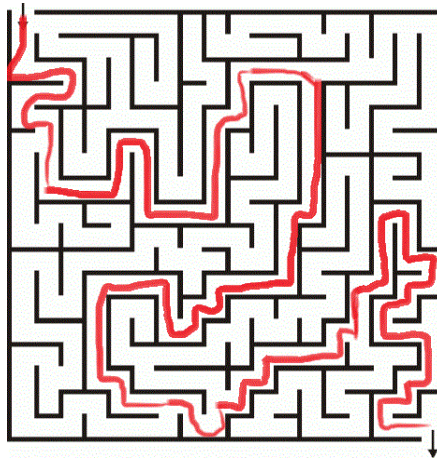
Pada subbab ini dijelaskan hasil pembuatan objek labirin. Dalam pembuatannya, pembuat mengikuti contoh dari gambar *maze* 2D yang didapat dari internet dengan tingkat kesulitan meningkat untuk pembuatan level 1 hingga level 3 permainan. Gambar *maze* yang dibuat menjadi contoh untuk dijadikan 3D, yang sebelumnya telah diubah/diberi tanda jalan, dapat dilihat pada Gambar 3.16, Gambar 3.17, dan Gambar 3.18.



Gambar 3.16 Model 2D untuk 3D labirin level 1, kiri untuk lantai 2 dan kanan untuk lantai 1

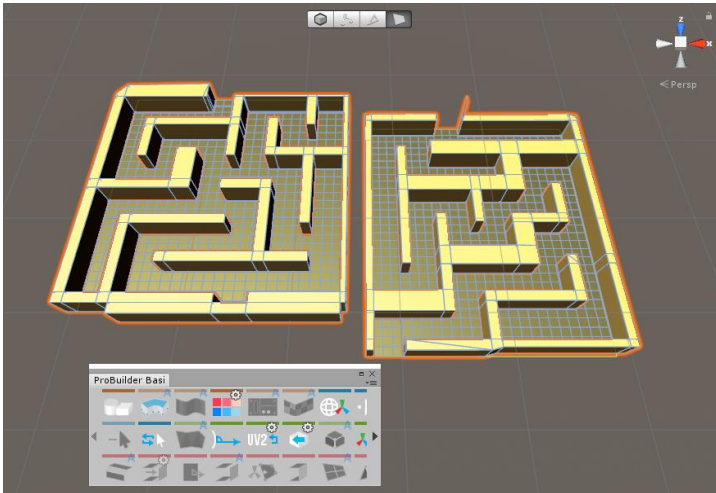


Gambar 3.17 Model 2D untuk 3D labirin level 2

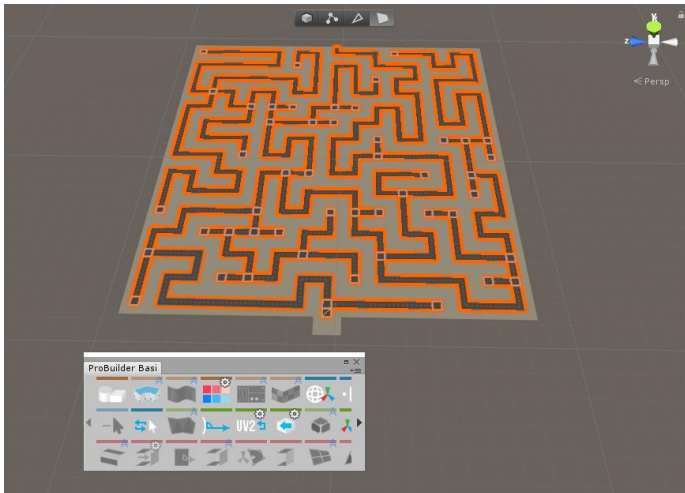


Gambar 3.18 Model 2D untuk 3D labirin level 3

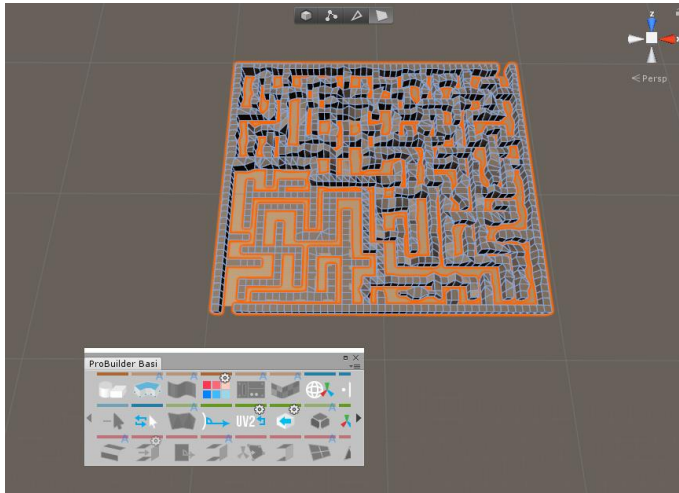
Kemudian untuk hasil model 3D dibuat dalam Unity dengan bantuan ProBuilder Basic dan dapat dilihat pada Gambar 3.19, Gambar 3.20, dan Gambar 3.21.



Gambar 3.19 Model 3D labirin level 1, kiri adalah lantai 2 dan kanan adalah lantai 1



Gambar 3.20 Model 3D labirin level 2



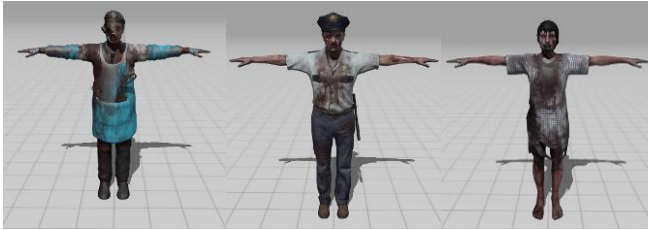
Gambar 3.21 Model 3D labirin level 3

3.2.6.8.2 Perancangan Model NPC

Pada subbab ini dijelaskan hasil pembuatan model NPC. Pada dasarnya, NPC dalam permainan ini dibuat menyerupai bentuk zombie, dibedakan menjadi musuh berjenis kelamin laki-laki dan perempuan. Musuh laki-laki memiliki kondisi lebih kuat namun bergerak lambat sedangkan musuh perempuan memiliki kondisi lebih lemah namun memiliki pergerakan lebih cepat. Model-model NPC ini adalah hasil modifikasi dari objek yang tersedia di situs mixamo.com. Perancangan objek NPC perempuan dapat dilihat pada Gambar 3.22 dan objek laki-laki pada Gambar 3.23.



Gambar 3.22 Objek NPC perempuan

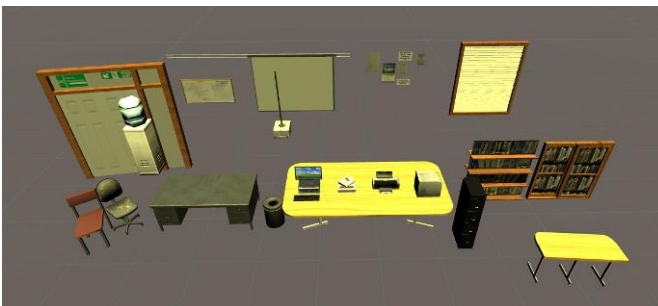


Gambar 3.23 Objek NPC laki-laki

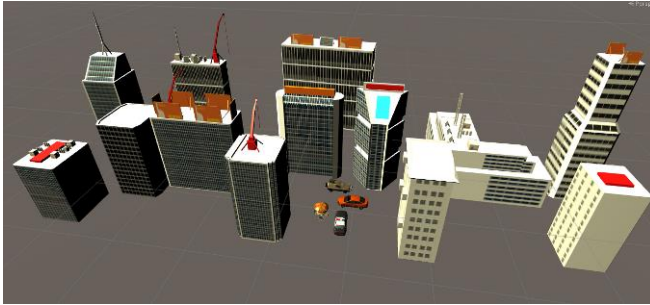
3.2.6.8.3 Perancangan Model-Model Pendukung Lain

Pada subbab ini dijelaskan pemakaian dan sebab penggunaan model-model 3D lain sebagai pendukung aplikasi. Model-model 3D didapat dari Asset Store Unity dengan mengunduh dan *import*.

Sesuai dengan isi dari bab 3.2.2 paragraf 2, maka pada level 1 permainan, lingkungan yang dibuat adalah lingkungan dalam ruangan kantor. Model-model pendukung untuk level 1 ditunjukkan pada Gambar 3.24. Pada level 2 permainan, lingkungan yang dibuat adalah lingkungan perkotaan, model-model pendukung untuk level 2 ditunjukkan pada Gambar 3.25. Pada level 3 permainan, lingkungan yang dibuat adalah lingkungan pedesaan, model-model pendukung untuk level 3 ditunjukkan pada Gambar 3.26.



Gambar 3.24 Objek-objek pendukung level 1

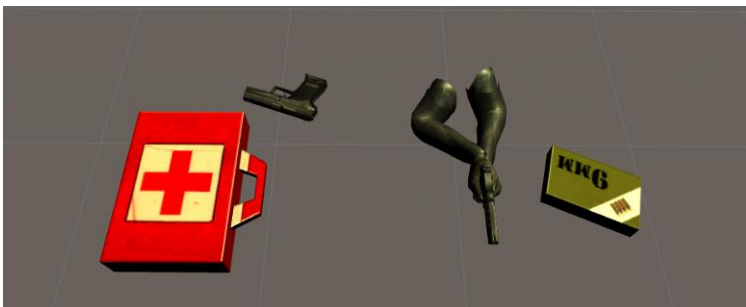


Gambar 3.25 Objek-objek pendukung level 2



Gambar 3.26 Objek-objek pendukung level 3

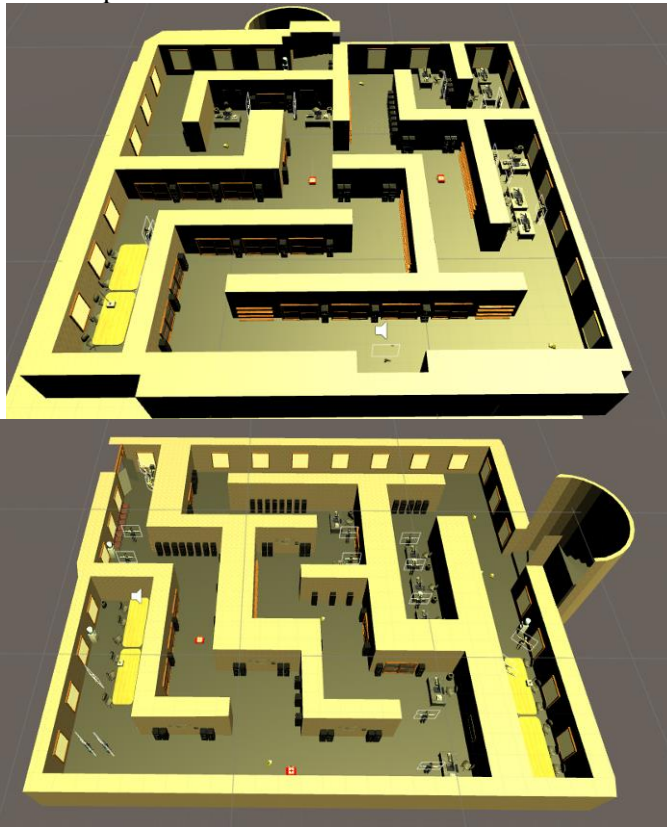
Selain model-model pendukung lingkungan level, diperlukan pula objek-objek untuk mendukung tampilan pemain. Model pendukung pemain ditunjukkan Gambar 3.27.



Gambar 3.27 Objek-objek pendukung tampilan pemain

3.2.6.8.4 Perancangan Desain Level

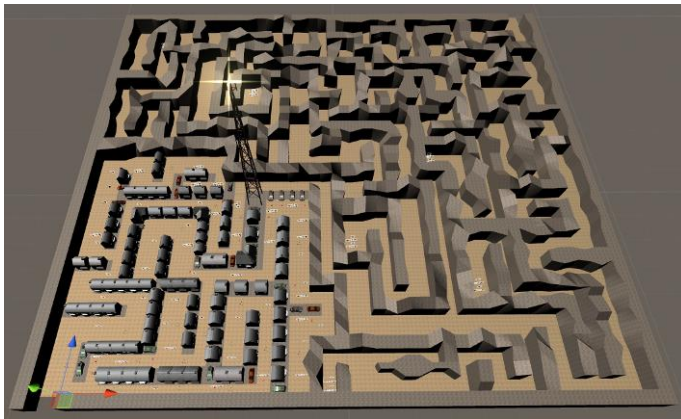
Pada subbab ini dijelaskan hasil dari pembuatan desain level untuk setiap level dalam permainan. Pembuatan desain level dibuat sesuai skenario bab 3.2.2 dengan model-model yang sudah dibuat sebelumnya. Pada level 1 permainan berlatarkan kantor, dan hasil dari desain level dapat dilihat pada Gambar. 3.28. Untuk level 2 permainan berlatarkan lingkungan perkotaan dan hasil dari desain level dapat dilihat pada Gambar 3.29. Sedangkan level 3 permainan berlatarkan pedesaan dan hasil dari desain levelnya dapat dilihat pada Gambar 3.30.



Gambar 3.28 Hasil desain level 1, lantai 2 (atas) dan lantai 1 (bawah)



Gambar 3.29 Hasil perancangan desain level 2



Gambar 3.30 Hasil perancangan desain level 3

3.2.6.9 Perancangan Kontrol Aplikasi

Pada aplikasi ini terdapat dua jenis perangkat input, yaitu *Smartphone* android. *Smartphone* akan membantu fungsi pergerakan arah tampilan aplikasi yang dilihat oleh pengguna. Apabila *smartphone* dimiringkan ke atas atau ke bawah, maka pergerakan kamera juga akan bergerak ke atas dan ke bawah.

Tidak hanya arah atas dan dan bawah, arah kamera juga akan mengikuti posisi *smartphone* sehingga dapat melihat ke arah 360 derajat.

Timer auto click, untuk input pada permainan, mengatur interaksi dengan tombol atau objek, pengguna dapat mengarahkan pointer atau yang dapat disebut *reticle* pada tengah layar ke arah tombol atau objek yang dituju dengan menjaganya sampai *timer* menunjukkan waktu habis dan melakukan suatu interaksi dengan objek yang diinginkan.

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian model, algoritma dan antarmuka yang mengacu pada rancangan yang telah dibahas pada bab sebelumnya.

4.1 Lingkungan Implementasi

Lingkungan implementasi tugas akhir dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat Keras	Pengembangan Aplikasi Prosesor : Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz Memori : 8 GB Implementasi Aplikasi Prosesor: Qualcomm Snapdragon 820 Quad core (2x2.15 GHz Kryo dan 2x1.6 GHz Kryo) Memori : 4 GB
Perangkat Lunak	Sistem Operasi : <ul style="list-style-type: none">- Windows 10 64 bit (Pengembangan)- Android 6.0.1 (Implementasi) Perangkat Pengembang : <ul style="list-style-type: none">- Unity 5.5.1f1- Visual Studio Community 2015 14.0.25431.01 Update3- Audacity 2.2.1

4.2 Implementasi Pembuatan Aplikasi

Pada tahap implementasi pembuatan aplikasi ini, akan dibagi ke dalam sembilan subbab. Kesembilan subbab tersebut yaitu meliputi Implementasi Antarmuka Menu, Implementasi Realitas Virtual dengan *Cardboard*, Implementasi Pembuatan *Scene*, Implementasi Import *Assets*, Implementasi *Load* Objek ke *Scene*, Implementasi Penambahan Audio, Implementasi Pembuatan *Script*, Implementasi Kontrol Aplikasi, dan Implementasi Mendapatkan Informasi.

4.2.1 Implementasi Antarmuka Menu

Antarmuka menu yang diimplementasikan oleh penulis antara lain antarmuka menu awal, antarmuka menu tutorial, dan antarmuka pilihan level. Antarmuka menu awal merupakan halaman yang ditampilkan ketika pengguna membuka aplikasi. Pada halaman ini terdapat tiga tombol yaitu tombol mulai, tutorial, dan keluar. Tombol mulai dan tombol tutorial akan membuka *scene* baru jika dipilih sedangkan tombol keluar akan menutup aplikasi. Untuk lebih jelasnya perhatikan Gambar 4.1. Sedangkan untuk mengatur tombol pergantian halaman dapat dilihat pada Kode Sumber 4.1.



Gambar 4.1 Antarmuka Halaman Menu

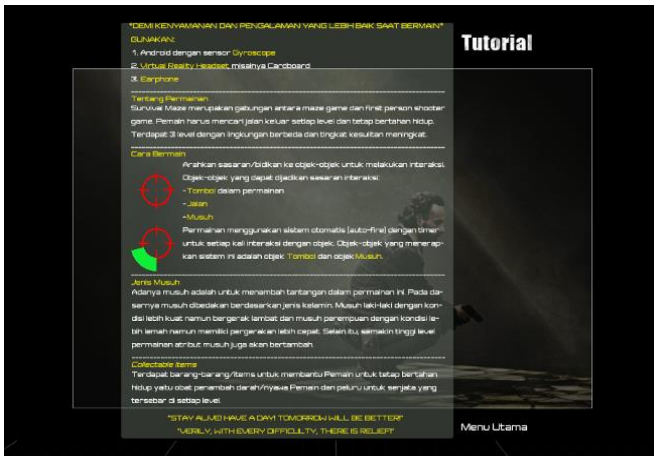

```
1. //Buttons
2. public void pLv()
3. {
4.     SceneManager.LoadScene("PilihLv");
5. }
6.
7. public void pTutorial()
8. {
9.     SceneManager.LoadScene("Tutorial");
10. }
11.
12. public void pQuit()
13. {
14.     Application.Quit();
15. }
```

Kode Sumber 4.1 Kontrol aksi tombol

Pada antarmuka tutorial berisi tentang tata cara bermain dan informasi tentang permainan, juga tombol untuk kembali ke menu utama. Tombol diatur oleh fungsi pada Kode Sumber 4.2. Tampilan antarmuka tutorial aplikasi dapat dilihat pada Gambar 4.2.

```
1. public void pHome()
2. {
3.     SceneManager.LoadScene("MENU");
4. }
```

Kode Sumber 4.2 Fungsi kembali ke menu utama



Gambar 4.2 Antarmuka tutorial

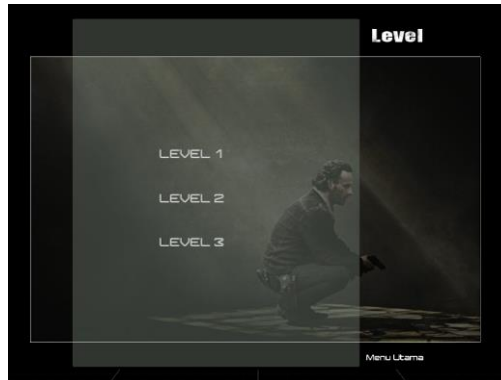
Saat menu mulai dipilih, akan menampilkan menu pilih level, dengan tombol yang diatur seperti Kode Sumber 4.3 dan antarmuka Gambar 4.3.

```

1. public void pLv1()
2. {
3.     SceneManager.LoadScene("lv1");
4. }
5.
6. public void pLv2()
7. {
8.     SceneManager.LoadScene("lv2");
9. }
10.
11. public void pLv3()
12. {
13.     SceneManager.LoadScene("lv3");
14. }

```

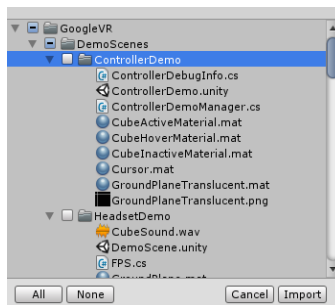
Kode Sumber 4.3 Fungsi memilih level



Gambar 4.3 Antarmuka pemilihan level

4.2.2 Implementasi Realitas Virtual dengan Cardboard

Untuk pembuatan proyek virtual reality pada perangkat bergerak, diperlukan SDK Google Carboard yang dapat diunduh pada <https://developers.google.com/vr>. Karena pengembangan aplikasi ini dilakukan pada Unity maka pilih Google VR SDK for Unity. Setelah SDK sudah diunduh maka import package kedalam proyek Unity. Caranya dengan memilih menu “Assets” lalu pilih “Import Package” dan “Custom Package”. Pilih SDK yang diunduh dan tunggu hingga proses load assets selesai. Jika sudah pilih tombol import. Perhatikan Gambar 4.4.



Gambar 4.4 Import Unity Package

4.2.3 Implementasi Pembuatan *Scene*

Untuk menambahkan scene, hal yang perlu dilakukan adalah menuju menu “File” lalu pilih “New Scene”. Setelah scene baru terbuat, kita dapat menambah game objek kedalam scene dan mengatur sesuai rancangan yang dibuat.

4.2.4 Implementasi Import Assets

Implementasi import assets dapat dilakukan dengan cara *drag and drop* dari file explore ke tab “Project” didalam folder “Assets” pada unity. Untuk lebih jelasnya dapat dilihat pada Gambar 4.5. Kita juga dapat mengatur pengelompokan asset dengan membuat folder-folder supaya lebih rapi dan memudahkan dalam pencarian asset.



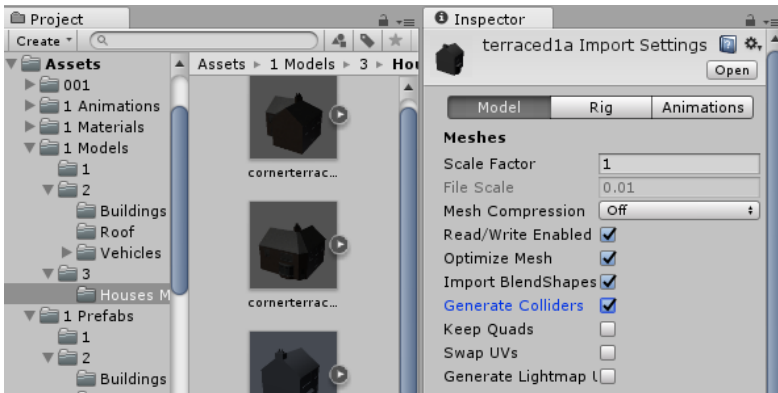
Gambar 4.5 Import Asset ke Proyek Unity

4.2.5 Implementasi *Load* Objek ke *Scene*

Untuk memasukan objek-objek ke dalam scene dapat dimasukkan dengan cara *drag and drop* berkas dari tab “Project” ke dalam *Scene*. Untuk melihat properti objek mengenai posisi atau yang lainnya, pilih objek yang akan dilihat propertinya. Informasi properti terdapat pada tab “Inspector”. Pada tab “Inspector” dapat ditambahkan beberapa komponen yang mempengaruhi objek pada *scene*.

Untuk objek 3D yang berasal dari file ekspor aplikasi blender dapat langsung ditambahkan *collider* agar dapat mendeteksi tumbukan antar objek. Hal ini diperlukan untuk mengatur pergerakan kamera agar tidak menembus objek.

Langkah-langkahnya yaitu pada aset berekstensi .fbx aktifkan *Generate Collider*, kemudian klik tombol *Apply* (Gambar 4.6).

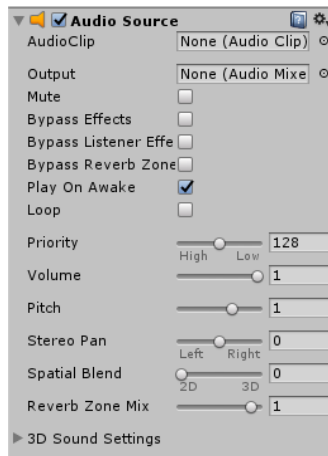


Gambar 4.6 *Generate Collider* pada Model 3D

Sedangkan untuk animasi yang berasal dari aplikasi blender dapat dilihat pada bagian “Animations”. Nama aplikasi dan pengaturan lainnya dapat juga diubah melalui bagian “Animations”.

4.2.6 Implementasi Penambahan Audio

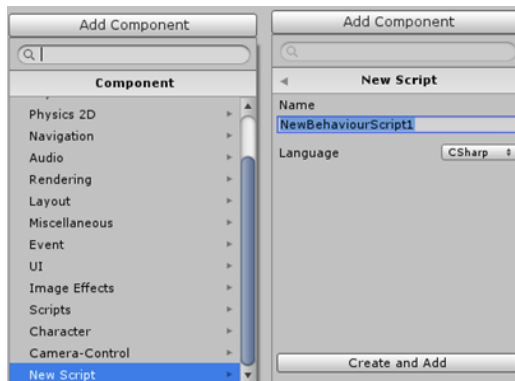
Untuk menambah audio langkah pertama yang harus dilakukan adalah pilih gameobject yang akan menyimpan audio tersebut. Kemudian pada tab “Inspector” tambahkan komponen *Audio Source*, jika berhasil maka akan muncul komponen *Audio Source* pada tab “Inspector” seperti pada Gambar 4.7. Pada komponen *Audio Source* terdapat *Audio Clip* yang berfungsi mengambil media music (.mp3 atau .wav) dari tab “Project”. Hilangkan centang pada *Play on awake* jika tidak ingin memutar audio saat *Load Scene* dan centang *Loop* jika ingin memutar kembali audio setelah mencapai akhir.



Gambar 4.7 Inspector Komponen Audio Source

4.2.7 Implementasi Pembuatan *Script*

Untuk membuat *script* yang mempengaruhi objek adalah dengan memilih terlebih dahulu objek yang akan diberi *script*. Pada tab “Inspector” tambahkan komponen *script*. Jika tidak ada *script* yang sesuai, kita akan membuat file *script* baru dan mengubah isi file sesuai yang kita inginkan (Gambar 4.8).



Gambar 4.8 Pembuatan *Script* Baru

4.2.8 Implementasi Kontrol Aplikasi

Kontrol aplikasi adalah suatu cara untuk interaksi pengguna dengan sistem. Pengembangan aplikasi realitas virtual yang menggunakan Google Cardboard mempunyai keterbatasan dalam implementasi kontrol, salah satunya adalah keterbatasan menerima input. Untuk itu dalam implementasi kontrol aplikasi ini penulis berusaha memberi alternatif dengan input berupa *trigger* otomatis dengan batas waktu untuk tiap kali melakukan *trigger*. Konsep yang penulis gunakan adalah dengan mendeteksi sebuah tombol dan memanfaatkan *Button script* yang sudah disediakan oleh Unity untuk melakukan suatu fungsi tertentu. Caranya adalah memberikan sebuah fungsi untuk mencari/mendeteksi sebuah tombol, kemudian menentukan apakah tombol tersebut masih tetap dipilih sampai waktu yang telah ditentukan, jika masih, maka akan menjalankan suatu fungsi yang sudah ditentukan/dibuat.

Pendeteksian tombol ini dapat menggunakan fitur *raycast* dan penandaan pada tombol. Penandaan dapat berupa nama atau *tag* atau apa saja sesuai keinginan pembuat program. Dalam kasus ini penulis menggunakan fitur *raycast* dan *tag*. Agar dapat berinteraksi dengan *Button script* dari Unity, maka kolom “Interactable” saat *Button script* dipanggil pada sebuah objek haruslah terpilih/tercentang. *Tag* yang dibuat oleh penulis digunakan untuk mengenali tombol atau musuh yang kemudian akan menjalankan interaksi tertentu sesuai fungsi yang sudah ditentukan. Untuk lebih jelasnya berikut adalah Kode Sumber 4.4 berikut.

```

1. using UnityEngine;
2. using UnityEngine.UI;
3. using System.Collections;
4.
5. namespace VRInteract {
6.     public class GazeRaycaster : MonoBehaviour {
7.         #region Public Variables
8.         public float loadingTime;
9.         public Image circle;

```



```

orm));
43.         return;
44.     } else {
45.         ResetGazer();
46.     }
47. } else {
48.     ResetGazer();
49. }
50. }
51. #endregion
52.
53. #region Private Methods
54. public IEnumerator FillCircle(Transform target)
55. {
56.     //Ketika mulai mengisi/loading, reset timer.
57.     float timer = 0f;
58.     circle.fillAmount = 0f;
59.     while (timer < loadingTime) {
60.         if (target.name != lastTargetName) {
61.             yield break;
62.         }
63.
64.         timer += Time.deltaTime;
65.         circle.fillAmount = timer / loadingTime;
66.
67.         yield return null;
68.     }
69.     circle.fillAmount = 1f;
70.
71.     if (target.GetComponent<Button>()) {
72.         target.GetComponent<Button>().onClick.Invoke();
73.     }
74.
75.     ResetGazer();
76. }
77.
78. // Reset loading nol dan ketika target image kosong
79. private void ResetGazer() {

```

```

80.         if (circle == null) {
81.             Debug.LogError("Please assign target load
ding image, (ie. circle image)");
82.             return;
83.         }
84.
85.         circle.fillAmount = 0f;
86.         lastTargetName = "";
87.     }
88.     #endregion
89. }
90. }

```

Kode Sumber 4.4 *Auto Click Invoke*

Langkah selanjutnya adalah menggunakan kode di atas untuk menghubungkan kode baru dengan *Button script* sehingga fungsi-fungsi baru yang diinginkan pada kode baru dapat terpanggil ketika ‘On Click()’, dengan syarat objek yang ingin diberikan fungsi tersebut harus memiliki tag ‘VRGazeInteractable’ atau ‘Enemy’.

4.2.8.1 Implementasi Jalan Otomatis

Pengguna dapat menelusuri tempat dengan cara berjalan otomatis, sehingga pengguna hanya perlu mengarahkan kamera ke arah objek yang memiliki tag/tanda “*path*” (objek lantai) untuk berjalan dan untuk berhenti pengguna hanya perlu mengarahkan kamera ke arah objek lain yang tidak memiliki tag/tanda “*path*”. Trigger tersebut akan mengubah nilai dari variable bertipe boolean “*walking*” (lihat Kode Sumber 4.5). Untuk itu diperlukan sebuah *script* yang mengatur gerakan kamera supaya dapat berjalan otomatis.

```

1.  if (walking) {
2.      transform.position = transform.position + Camera.main.transform.forward * speed * Time.deltaTime;
3.  }
4.  Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5f, 0.5f, 0));
5.  RaycastHit hit;

```

```

6.
7.     if (Physics.Raycast(ray, out hit)) {
8.         if (hit.collider.tag.Contains("path")) {
9.             walking = true;
10.        }
11.        else{
12.            walking = false;
13.        }
14.    }

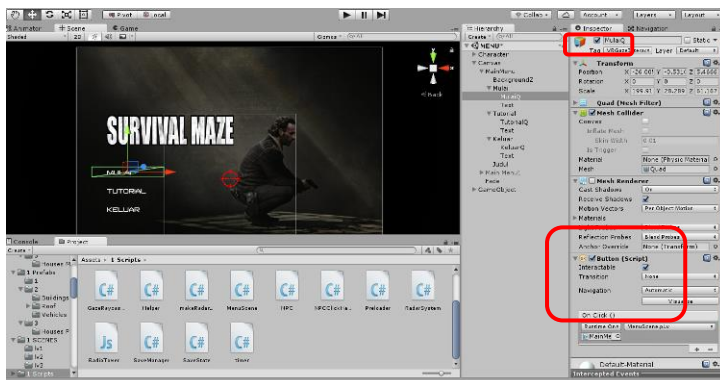
```

Kode Sumber 4.5 Jalan Otomatis

Pada Kode Sumber 4.5 terdapat variable “head” yang bertipe *GvrHead* (kelas dari package SDK Google VR). Variable digunakan untuk mengambil arah dari kamera. Terdapat juga variabel “speed” yang mengatur kecepatan jalan kamera/pemain.

4.2.8.2 Implementasi Pilih Level

Pada pemilihan level, pengguna perlu memberi *trigger on click button* pada tombol yang dilihat (dalam hal ini dibantu sebuah target titik pada tengah kamera (*reticle*) untuk konfirmasi pilihan. Untuk membuat kode program berjalan maka diperlukan untuk menambah komponen *Collider* pada masing-masing menu tombol (Gambar 4.9).

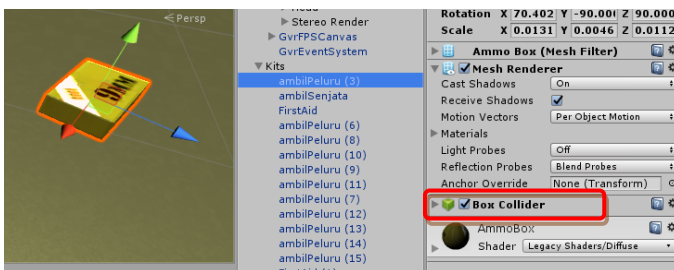


Gambar 4.9 Pengaturan Komponen pada Tombol

4.2.9 Implementasi Mendapatkan Informasi

Untuk menampilkan sebuah informasi maka diperlukan sebuah *trigger* yang akan menampilkan informasi pada *screen* android jika *trigger* tersebut aktif. Hal yang diperlukan adalah sebuah canvas yang menampung isi informasi dan kotak *trigger* yang akan menampilkan informasi. Langkah-langkahnya adalah sebagai berikut:

1. Buat objek untuk interaksi, pastikan sudah memiliki collider, selanjutnya membuat fungsi `OnCollisionEnter()` untuk *player*. Dalam permainan ini terdapat tiga jenis barang bantuan, yaitu senjata, amunisi, dan kotak obat untuk *player*. Untuk lebih jelasnya lihat contoh Gambar 4.10 dan fungsinya pada Kode Sumber 4.6
2. Tambahkan canvas pada hirarki proyek yang di dalamnya memuat latar informasi dan teks yang berisi informasi (Gambar 4.11). Atur mode “Render Canvas” sebagai *world space* dan *event camera* merujuk pada Main Camera di hirarki proyek.
3. Pada latar informasi, tambahkan kode program yang mengatur informasi supaya selalu tampil mengikuti arah kamera. Perhatikan Gambar 4.12.
4. Jika menginginkan canvas mengikuti suatu objek berjalan maka masukkan canvas ke dalam *child* objek tersebut, terutama jika objek memiliki kepala (Gambar 4.13)



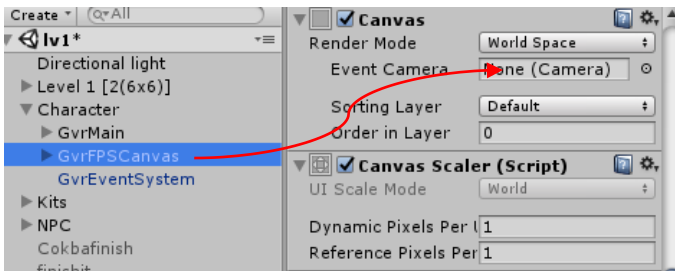
Gambar 4.10 Membuat Kotak atau Barang Bantuan

```

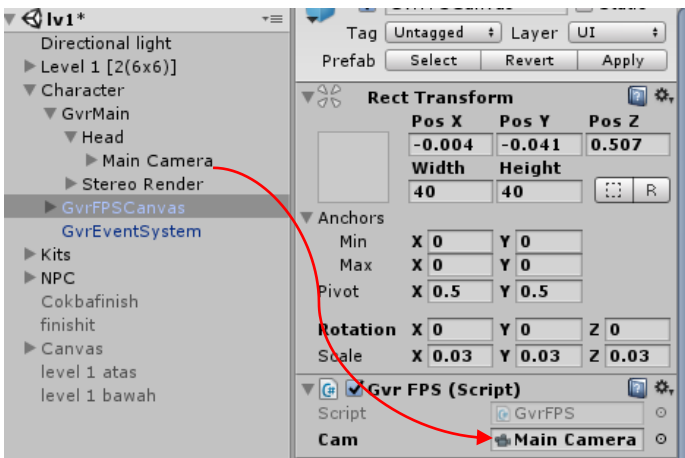
1. void OnCollisionEnter(Collision hit) {
2.     //Ambil senjata
3.     if (hit.gameObject.name == "ambilSenjata")
4.     {
5.         gun.gameObject.SetActive(true);
6.         fps_shoot.instance.hasGun += 1;
7.         Destroy(hit.gameObject);
8.         audioSource = GetComponent<AudioSource>
9.         ();
10.        audioSource.clip = ambilSenjata;
11.        audioSource.Play();
12.    }
13.    //Ambil peluru +7
14.    if (hit.gameObject.tag == "peluru") {
15.        fps_shoot.instance.ammo += 7;
16.        Destroy(hit.gameObject);
17.        audioSource = GetComponent<AudioSource>
18.        ();
19.        audioSource.clip = ambilPeluru;
20.        audioSource.Play();
21.    }
22.    //Ambil obat
23.    if (hit.gameObject.tag == "potion") {
24.        playerHealth += 50;
25.        Destroy(hit.gameObject);
26.        audioSource = GetComponent<AudioSource>
27.        ();
28.        audioSource.clip = ambilPotion;
29.        audioSource.Play();
30.    }

```

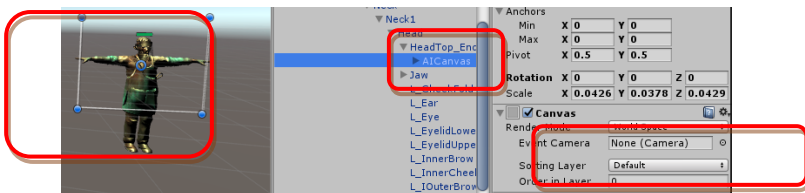
Kode Sumber 4.6 Ketika menabrak Barang Bantuan



Gambar 4.11 Membuat Area Informasi



Gambar 4.12 Mengatur posisi teks informasi terhadap kamera



Gambar 4.13 Mengatur posisi teks informasi terhadap objek bergerak

4.3 Implementasi Aplikasi Permainan

Dalam implementasi permainan akan dibagi menjadi empat subbab yang meliputi implementasi *finite state machine* karakter NPC, implementasi menelusuri labirin dan mengambil barang bantuan, implementasi sistem FPS, serta implementasi menemukan *finish*.

4.3.1 Implementasi Finite State Machine Karakter NPC

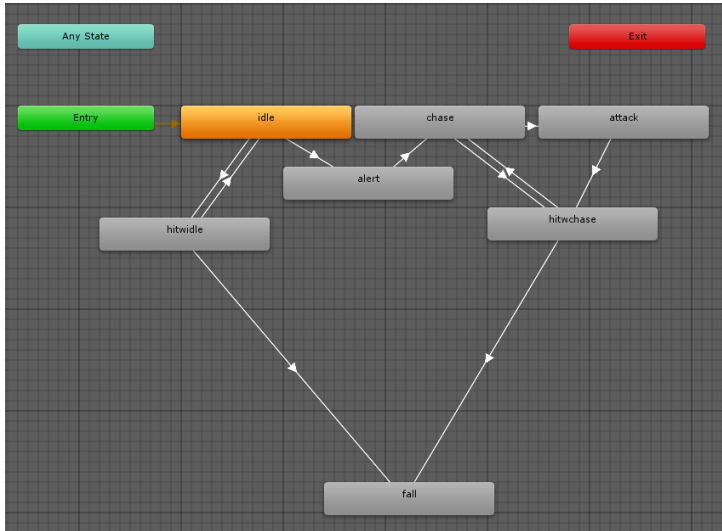
Finite State Machine (FSM) adalah konseptual mesin yang hanya bisa menjalankan satu dari sekian *finite state* dalam satu waktu. FSM biasanya digunakan dalam bidang kecerdasan buatan untuk menentukan perilaku dari *computer-controlled character*, dalam hal ini adalah NPC/musuh.

Unity menyediakan *mechanim system* yang menggunakan FSM untuk mengontrol perilaku animasi. Maka dari itu, penulis menggunakan fitur dalam Unity, yaitu Animator Controller untuk memberikan pengaturan perilaku NPC/musuh yang telah diberi animasi sebelumnya. Dalam animator controller. Keadaan-keadaan utama untuk musuh dibagi menjadi empat, yaitu keadaan ketika diam (*idle*), keadaan mengejar pemain, keadaan menyerang pemain, serta keadaan musuh mati. Setiap keadaan tersebut memiliki animasi dan penulis juga mnyertakan tambahan animasi untuk membuat perpindahan animasi lebih halus.

Kondisi utama NPC adalah diam (*idle*), saat kondisi ini, syaratnya adalah NPC masih hidup. Ketika *player* menembak NPC yang dalam keadaan ‘idle’, akan menjadi *trigger* kondisi ‘hitwidle’ aktif dan menjalankan animasi di dalam kondisi tersebut, kemudian kembali lagi menjadi ‘idle’. Namun ketika posisi *player* sudah masuk jarak pandang NPC, maka kondisi ‘idle’ menjadi *false* dan kondisi ‘alert’ menjadi *true*.

Setelah kondisi ‘chase’ menjadi *true*, maka kondisi ‘alert’ dijadikan *false*. Dari kondisi ‘chase’ akan berpindah kepada ‘attack’ ketika parameter ‘attack’ menjadi *true*. Namun, baik dari kondisi ‘chase’ atau ‘attack’, akan terjadi *trigger* kondisi ‘hitwchase’ saat *player* menembakkan senjatanya, setelah kondisi

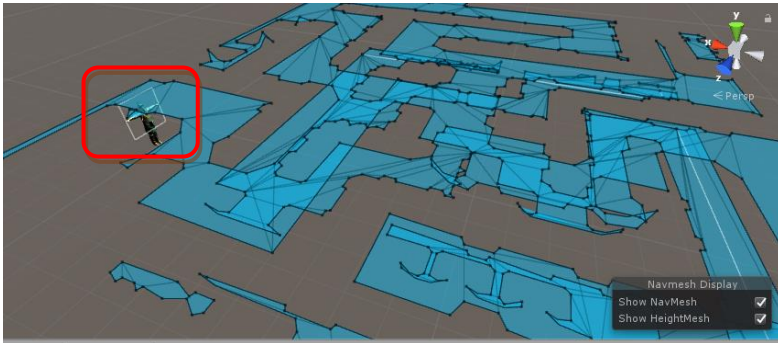
ini terjadi akan langsung kembali ke kondisi ‘chase’. Bagian terakhir adalah baik dari kondisi ‘hitwidle’ maupun ‘hitwchase’ dapat menjadi kondisi ‘fall’ ketika parameternya sesuai. Untuk lebih jelasnya dapat melihat Gambar 4.14



Gambar 4.14 Pengaturan *Animator Controller*

Ketika sudah selesai membuat FSM dengan Animator Controller, yang selanjutnya harus dilakukan adalah menentukan jalan yang dapat dilalui dan tidak dapat dilalui oleh NPC dengan Navigation. Navigation dalam Unity dapat mengatur area mana yang dapat dilalui dan tidak dapat dilalui oleh suatu objek, terutama objek dengan atribut NavMesh Agent.

NavMesh Agent digunakan untuk membuat karakter yang diberi atribut dapat mencapai tujuannya (*goal position*), dalam hal ini *player*, dengan tetap melewati navigation area, sehingga tidak akan terjadi karakter NPC yang menembus dinding atau semacam itu. Untuk lebih jelasnya pada Gambar 4.15



Gambar 4.15 Navigation Area untuk Level 1

Pada gambar di atas, terdapat satu NPC yang telah diberi NavMesh Agent dan sebuah Navigation Area. Yang akan terjadi selanjutnya adalah NPC tersebut akan mencari tujuannya dengan menjalankan semua variabel NavMesh Agent.

Untuk menggabungkan semua itu diperlukan sebuah kode program untuk NPC, yang ditunjukkan pada Kode Sumber 4.7.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class NPC : MonoBehaviour {
7.     private Animator anim;
8.     private UnityEngine.AI.NavMeshAgent agent;
9.     private bool isDead = false;
10.    public Transform player;
11.    public float enemymaxHealth = 30f;
12.    public float enemyHealth;
13.    public float damage = 10f;
14.    public GameObject gun;
15.    public GameObject gunObject;
16.    public fps_shoot script;
17.    public Image Fill;
18.    public float damping;
19.    public float distance;
20.    public bool valklik;
21.

```

```

22. public void Start()
23. {
24.     valklik = true;
25.     script = gunObject.GetComponent<fps_shoot>();
26.     anim = GetComponent<Animator>();
27.     enemyHealth = enemymaxHealth;
28.     agent = gameObject.GetComponent<UnityEngine.AI.
        NavMeshAgent>();
29. }
30.
31. void Update()
32. {
33.     if (!isDead) {
34.         if (enemyHealth <= 0f) {
35.             Destroy(GetComponent<BoxCollider>());
36.             Destroy(GetComponent<SphereCollider>())
37.             ;
38.             gameObject.tag = "Untagged";
39.             enemyHealth = 0f;
40.             isDead = true;
41.             anim.SetBool("isFall", true);
42.             anim.SetBool("isIdle", false);
43.             anim.SetBool("isAlert", false);
44.             anim.SetBool("isChase", false);
45.             anim.SetBool("isAtt1", false);
46.             agent.SetDestination(agent.transform.po
        sition);
47.             Destroy(gameObject,10);
48.         }
49.         else {
50.             distance = Vector3.Distance(player.trans
        form.position, agent.transform.position);
51.             //alert
52.             if (distance < 11) {
53.                 lookAtPlayer();
54.                 anim.SetBool("isIdle", false);
55.                 anim.SetBool("isAlert", true);
56.                 anim.SetBool("isFall", false);
57.                 anim.SetBool("isAtt1", false);
58.             } anim.SetBool("isChase", true);
59.             //attack
60.             if (distance <= 2) {
                GameObject.Find("GvrMain").GetCompo

```

```

    nent<fps_script>().playerHealth -
    = damage * Time.deltaTime;
61.         anim.SetBool("isAlert", false);
62.         anim.SetBool("isAtt1", true);
63.         anim.SetBool("isFall", false);
64.         anim.SetBool("isIdle", false);
65.         anim.SetBool("isChase", false);
66.         if (!isDead)
67.         {
68.             agent.SetDestination(agent.trans
form.position);
69.         }
70.         else anim.SetBool("isAtt1", fals
e);
71.     }
72.     //chase
73.     else if (distance < 10.5) {
74.         agent.SetDestination(player.transfo
rm.position);
75.         if (distance > 2) {
76.             anim.SetBool("isChase", true);
77.             anim.SetBool("isAlert", false);
78.             anim.SetBool("isAtt1", false);
79.             anim.SetBool("isFall", false);
80.             anim.SetBool("isIdle", false);
81.         }
82.     }
83. }
84. }
85. }
86.
87. void lookAtPlayer()
88. {
89.     Quaternion rotation = Quaternion.LookRotation(p
layer.position - transform.position);
90.     transform.rotation = Quaternion.Slerp(transform
.rotation, rotation, Time.deltaTime * damping);
91. }

```

```

92.
93. public void enemyManager()
94. {
95.     if (gun.gameObject.activeInHierarchy)
96.     {
97.         script.shot();
98.         anim.SetTrigger("tryTrigger");
99.         //anim.SetBool("isHitwchase", true);
100.         enemyHealth -= damage;
101.         float calc_health = enemyHealth / enemymaxHealth;
102.         enemycurHealth(calc_health);
103.         //print(enemyHealth);
104.     }
105. }
106.
107. void enemycurHealth(float aiHealth) {
108.     Fill.fillAmount = aiHealth;
109. }
110. }

```

Kode Sumber 4.7 Kode Program untuk NPC

4.3.2 Implementasi Menelusuri Labirin dan Mengambil Barang Bantuan

Dalam implementasi aturan permainan pada aplikasi ini pengguna diminta untuk tetap bertahan hidup dengan mengumpulkan barang bantuan yang ditentukan di jalan ketika mencari jalan keluar dari suatu level. Untuk mengambil barang dalam permainan tersebut pengguna harus menabrak objek koin.

Untuk pembuatan proses mengambil nilai dari barang yang diambil diperlukan kode program sesuai Kode Sumber 4.8 berikut.

```

29. void OnCollisionEnter(Collision hit) {
30.     //Ambil senjata
31.     if (hit.gameObject.name == "ambilSenjata") {
32.         gun.gameObject.SetActive(true);
33.         fps_shoot.instance.hasGun += 1;
34.         Destroy(hit.gameObject);
35.         audioSource = GetComponent();

```

```

36.         audioSource.clip = ambilSenjata;
37.         audioSource.Play();
38.     }
39.
40.     //Ambil peluru +7
41.     if (hit.gameObject.tag == "peluru") {
42.         fps_shoot.instance.ammo += 7;
43.         Destroy(hit.gameObject);
44.         audioSource = GetComponent();
45.
46.         audioSource.clip = ambilPeluru;
47.         audioSource.Play();
48.     }
49.
50.     //Ambil obat
51.     if (hit.gameObject.tag == "potion") {
52.         playerHealth += 50;
53.         Destroy(hit.gameObject);
54.         audioSource = GetComponent();
55.
56.         audioSource.clip = ambilPotion;
57.         audioSource.Play();
58.     }

```

Kode Sumber 4.8 Implementasi Mengambil Nilai Barang Bantuan

4.3.3 Implementasi First Person Shooter

Implementasi untuk FPS membutuhkan kode program untuk mendapat informasi dengan sudut pandang orang pertama dan kode program untuk melakukan interaksi, dalam hal ini adalah menembak musuh. Kode program dapat dilihat pada Kode Sumber 4.9 dan 4.10.

```

1. using UnityEngine;
2. using UnityEngine.UI;
3. using System.Collections;
4. using System.Collections.Generic;
5.
6. public class fps_script : MonoBehaviour {
7.
8.     public static fps_script instance;

```

```

9.     public float playermaxHealth = 300;
10.    public float playerHealth;
11.    public bool walking = false;
12.    public float speed = 0.9f;
13.    public int finish = 0;
14.    float timer;
15.    public TextMesh health_text;
16.    public TextMesh finish_text;
17.    public Image Fill;
18.    public AudioSource audioSource;
19.    public AudioClip ambilPeluru;
20.    public AudioClip ambilPotion;
21.    public AudioClip ambilSenjata;
22.    public AudioClip painSound;
23.    public GameObject gun;
24.
25.    void Start() {
26.        instance = this;
27.        playerHealth = playermaxHealth;
28.        audioSource = gameObject.AddComponent<AudioSource>();
29.    }
30.
31.
32.    void Update() {
33.
34.        if (walking) {
35.            transform.position = transform.position + Camera.main.transform.forward * speed * Time.deltaTime;
36.        }
37.        Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5f, 0.5f, 0));
38.        RaycastHit hit;
39.
40.        if (Physics.Raycast(ray, out hit)) {
41.            if (hit.collider.tag.Contains("path")) {
42.                walking = true;
43.            }
44.            else{
45.                walking = false;
46.            }
47.        }

```

```

48.
49.     timer += Time.deltaTime;
50.
51.     //Attacked
52.     float calc_health = playerHealth / playermaxHea
    lth;
53.     playercurHealth(calc_health);
54.     if (playerHealth <= 0f)
55.     {
56.         health_text.text = "KAMU MATII!";
57.     }
58.
59.     if (Input.GetKeyDown(KeyCode.P)) {
60.         if (Time.timeScale == 1) {
61.             Time.timeScale = 0;
62.         }
63.     else {
64.         Time.timeScale = 1;
65.     }
66.     }
67. }

```

Kode Sumber 4.9 Implementasi Kode Program *Player*

```

1. using UnityEngine;
2. using System.Collections;
3. using UnityEngine.UI;
4.
5. public class fps_shoot : MonoBehaviour {
6.     private AudioSource audioSource;
7.     public static fps_shoot instance;
8.     public Rigidbody projectile;
9.     public float speed = 20;
10.    public int hasGun = 0;
11.    public int ammo = 7;
12.    public float ammomaxing=7;
13.    public float ammoimg;
14.    public AudioClip reload;
15.    public AudioClip Scored;
16.    public Image Fill;
17.    public TextMesh ammo_text;
18.

```

```

19.     // Use this for initialization
20.     void Start () {
21.         instance = this; //this is so all public variables
           can be accessed outside
22.         ammoimg = ammomaximg;
23.         ammo_text.text = '\t' + "           : " + ammo.
           ToString();
24.     }
25.
26.     // Update is called once per frame
27.     void Update () {
28.
29.
30.         if (ammo >0)
31.         {
32.             ammo_text.text = '\t' + "           : " + a
           mmo.ToString();
33.         }
34.         else ammo_text.text = '\n' + "PELURU HABIS!";
35.     }
36.
37.     public void shot() {
38.         if (hasGun == 1 && ammo > 0)
39.         {
40.             ammo -= 1;
41.             Rigidbody bullet = Instantiate(projectile, tra
           nsform.position, transform.rotation) as Rigidbody;
42.             bullet.velocity = transform.TransformDirection
           (new Vector3(0, 0, speed));
43.             Destroy(bullet, 2);
44.             audioSource = GetComponent<AudioSource>();
45.             audioSource.clip = Scored;
46.             audioSource.Play(); //play explosion sound
47.             ammoimg -= 1;
48.             float calc_ammo = ammoimg / ammomaximg;
49.             ammocuring(calc_ammo);
50.         }
51.     }
52.
53.     void ammocuring(float ammocur){
54.         Fill.fillAmount = ammocur;
55.         if (ammocur <= 0 && ammo >0) {
56.             //animas();

```



```

57.         audioSource = GetComponent();
58.         audioSource.clip = reload;
59.         audioSource.Play();
60.         ammoing = ammomaxing;
61.         Fill.fillAmount = 1;
62.     }
63. }
64. }

```

Kode Sumber 4.10 Implementasi Kode Program *Player*

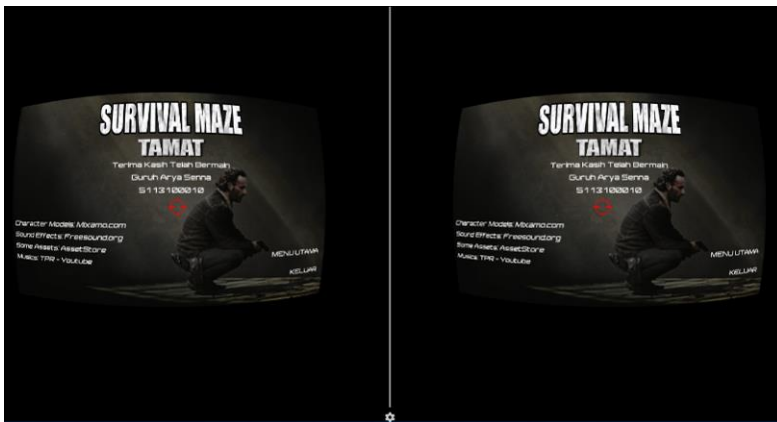
4.3.4 Implementasi Menemukan Finish

Pada saat pengguna telah menemukan garis *finish* dari labirin maka permainan pada *level* tersebut telah selesai. Pengguna akan mendapatkan pilihan untuk melanjutkan ke *level* selanjutnya. Tampilan pilihan saat mencapai finish dapat dilihat pada Gambar 4.16. Pilihan menu tersebut hanya tampil saat pengguna mencapai *finish*. Untuk membuat menu tersebut langkah pertama adalah memberikan collider dan trigger pada objek *finish*. Kemudian cek trigger, jika aktif maka tampilkan menu.

Apabila *level* 3 telah selesai/mencapai *finish* dan pemain memilih *level* selanjutnya, permainan akan menampilkan *scene* lain yaitu yang menandakan telah menamatkan permainan ini. Tampilan tamat permainan dapat dilihat pada Gambar 4.17.



Gambar 4.16 Tampilan menu saat game *finish*



Gambar 4.17 Tampilan menu saat game *finish*

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini membahas mengenai uji coba dan evaluasi aplikasi “Survival Maze” pada perangkat bergerak dengan teknologi Google Cardboard. Uji coba dilakukan menggunakan metode *black box* (kotak hitam), yaitu pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak berdasarkan skenario yang telah ditentukan.

5.1 Lingkungan Uji Coba

Pada proses uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan perangkat lunak. Berikut ini akan dijelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1. Uji coba ini dilakukan menggunakan kacamata VR Box dengan *earphone* sebagai alat bantu dengar untuk pengguna.

Tabel 5.1 Lingkungan Perangkat Keras

No.	Deskripsi
1	Model Perangkat Samsung Galaxy S7 edge Chipset : Qualcomm Snapdragon 820 CPU: Quad-core (2x2.15 GHz Kryo & 2x1.6 GHz Kryo) Memori : 4.00 GB Sistem Operasi : Android 6.0.1 API level: 23
3	Model Perangkat Xiaomi Mi 5 Chipset : Qualcomm Snapdragon 820 CPU : Quad-core (2x2.18 GHz Kryo & 2x1.36 GHz Kryo) Memori : 4.00 GB Sistem Operasi : Android 6.0.1 API level : 23

5.2 Skenario dan Hasil Uji Coba

Pada subbab ini dijelaskan mengenai skenario yang dilakukan untuk mendapatkan hasil uji coba aplikasi. Skenario uji coba aplikasi dilakukan untuk mengetahui ketercapaian aplikasi dalam memenuhi kebutuhan fungsionalitas dan kebutuhan non-fungsionalitas.

5.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas aplikasi ini dapat dilakukan secara mandiri. Pengujian ini bertujuan untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan. Skenario yang dibuat mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.5.2. Skenario uji coba fungsionalitas yang dilakukan terhadap aplikasi yang dibangun dijelaskan pada Tabel 5.2.

Tabel 5.2 Skenario Uji Coba Fungsionalitas

Kode Uji Coba	Nama Uji Coba
UJ-UC-001	Uji Memilih Level
UJ-UC-002	Uji Menelusuri Labirin
UJ-UC-003	Uji Melihat Informasi Atribut Player dan NPC
UJ-UC-004	Uji Menemukan Finish

Setiap skenario akan dijelaskan mengenai kondisi awal, masukkan, dan keluaran yang diharapkan, kondisi akhir, dan hasil uji coba. Berikut ini merupakan penjabaran hasil setiap uji coba yang dilakukan.

5.2.1.1 Uji Memilih Level

Uji coba ini bertujuan untuk membawa pengguna memasuki level yang ingin ditelusuri. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil Uji Coba Memilih Level

Kode Uji Coba	UJ-UC-001
Kondisi Awal	Aplikasi menampilkan tiga pilihan level
Skenario 1 (Pengguna mengarahkan reticle ke pilihan stage)	
Masukan	Mengarahkan <i>reticle</i> ke kotak pilihan level
Keluaran yang diharapkan	Muncul timer untuk memilih menu
Kondisi Akhir	Timer habis dan berganti scene
Hasil Uji Coba	Berhasil
Skenario 2 (Pengguna mengarahkan reticle ke menu utama)	
Masukan	Mengarahkan <i>reticle</i> ke menu utama
Keluaran yang diharapkan	Muncul timer untuk memilih menu
Kondisi Akhir	Timer habis dan berganti scene menu utama
Hasil Uji Coba	Berhasil

5.2.1.2 Uji Menelusuri Labirin

Uji coba ini bertujuan untuk mengarahkan pengguna berkeliling dalam realitas virtual labirin. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Uji Coba Menelusuri Tempat

Kode Uji Coba	UJ-UC-002
Kondisi Awal	Proses <i>load</i> objek sudah selesai
Skenario 1 (Pengguna mengarahkan <i>smartphone</i> ke atas, bawah, kanan, kiri, dan belakang)	
Masukan	Mengarahkan <i>smartphone</i> ke sembarang arah
Keluaran yang diharapkan	Layar <i>smartphone</i> menampilkan objek 3D sesuai arah <i>smartphone</i>
Kondisi Akhir	Layar <i>smartphone</i> menampilkan objek 3D sesuai arah <i>smartphone</i>

Hasil Uji Coba	Berhasil
Skenario 2 <i>(Pengguna mengarahkan smartphone kebawah dalam kondisi diam)</i>	
Masukan	Mengarahkan kamera ke arah lantai
Keluaran yang diharapkan	Kamera bergerak ke depan
Kondisi Akhir	Kamera bergerak ke depan
Hasil Uji Coba	Berhasil
Skenario 3 <i>(Pengguna mengarahkan smartphone keatas dalam kondisi bergerak)</i>	
Masukan	Mengarahkan kamera ke arah selain lantai
Keluaran yang diharapkan	Kamera berhenti
Kondisi Akhir	Kamera berhenti
Hasil Uji Coba	Berhasil
Skenario 4 <i>(Pengguna berhenti hingga nyawa habis dan memulai ulang scene stage sekarang)</i>	
Masukan	Mengarahkan kamera selain ke arah lantai sehingga berhenti
Keluaran yang diharapkan	Nyawa habis, menampilkan <i>game over scene</i> , dapat mengulang <i>scene stage</i> sekarang
Kondisi Akhir	Memulai ulang <i>scene stage</i> sekarang
Hasil Uji Coba	Berhasil

5.2.1.3 Uji Memperoleh Informasi Atribut Player dan NPC

Uji coba ini bertujuan untuk memberi pengguna informasi mengenai objek-objek serta atribut yang dibawanya. Skenario dan hasil pengujian secara lengkap dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil Uji Coba Melihat Informasi

Kode Uji Coba	UJ-UC-003
Kondisi Awal	Proses <i>load</i> objek sudah selesai
Skenario 1 (Pengguna menabrak objek berupa kotak peluru atau kotak obat)	
Masukan	Pengguna menabrak objek kotak peluru atau kotak obat
Keluaran yang diharapkan	Layar <i>smartphone</i> menampilkan perubahan informasi teks pada jumlah sisa peluru atau jumlah sisa nyawa pemain.
Kondisi Akhir	Layar <i>smartphone</i> menampilkan perubahan informasi nilai pada amunisi atau nyawa
Hasil Uji Coba	Berhasil
Skenario 2 (Pengguna menelusuri labirin, waktu tetap berjalan)	
Masukan	Mengarahkan <i>smartphone</i> ke sembarang arah
Keluaran yang diharapkan	Layar <i>smartphone</i> menampilkan perubahan informasi teks pada waktu
Kondisi Akhir	Layar <i>smartphone</i> menampilkan perubahan informasi teks pada waktu
Hasil Uji Coba	Berhasil
Skenario 3 (Pengguna menelusuri labirin, memilih untuk kembali ke menu sebelumnya)	
Masukan	Mengarahkan <i>smartphone</i> ke arah atas pemain
Keluaran yang diharapkan	Memilih tombol kembali kemudian membuka <i>scene</i> pilih level.
Kondisi Akhir	Membuka <i>scene</i> pilih level.
Hasil Uji Coba	Berhasil

5.2.1.4 Uji Menemukan Finish

Uji coba ini bertujuan untuk memberi pengguna informasi pilihan menu untuk menuju ke stage berikutnya. Skenario dan hasil dapat dilihat pada Tabel 5.6.

Tabel 5.6 Hasil Uji Coba Menemukan Finish

Kode Uji Coba	UJ-UC-004
Kondisi Awal	Proses <i>load</i> objek sudah selesai
Skenario 1 (Pengguna berjalan menelusuri labirin)	
Masukan	Pengguna menabrak objek <i>finish</i>
Keluaran yang diharapkan	Layar <i>smartphone</i> menampilkan informasi pilhan menu
Kondisi Akhir	Layar <i>smartphone</i> menampilkan informasi pilhan menu
Hasil Uji Coba	Berhasil
Skenario 2 (Pengguna memilih salah satu pilihan menu)	
Masukan	Memilih salah satu dari pilihan menu yang ditampilkan
Keluaran yang diharapkan	Masuk ke <i>scene</i> sesuai menu yang telah dipilih
Kondisi Akhir	Masuk ke <i>scene</i> sesuai menu yang telah dipilih
Hasil Uji Coba	Berhasil

5.2.2 Pengujian Pengguna

Pengujian pada *game* yang dibangun tidak hanya dilakukan pada fungsionalitas yang dimiliki, tetapi juga pada pengguna untuk melakukan percobaan secara langsung. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan *game* yang dibangun dari sisi pengguna. Hal ini dapat dicapai dengan meminta penilaian dan tanggapan dari pengguna terhadap sejumlah aspek perangkat lunak yang ada.

5.2.2.1 Skenario Uji Coba Pengguna

Dalam melakukan pengujian aplikasi ini, penguji diminta untuk mencoba menggunakan aplikasi yang bersangkutan dan mencoba semua fungsionalitas serta fitur yang tersedia. Selain itu, pengguna juga diminta memberikan saran untuk pengembangan aplikasi selanjutnya.

Pengujian aplikasi oleh pengguna dilakukan dengan sebelumnya memberi informasi seputar aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba aplikasi dengan spesifikasi lingkungan yang sama persis dengan yang telah diuraikan pada uji coba fungsionalitas.

Jumlah pengguna yang terlibat dalam pengujian aplikasi ini sebanyak sebelas orang, dengan rincian 6 orang mahasiswa Teknik Informatika dan sisanya bukan dari Teknik Informatika. Dalam melakukan pengujian, pengguna melakukan percobaan sebanyak satu kali penggunaan.

Ketika memberikan penilaian dan tanggapan terhadap aplikasi, penguji diberikan formulir pengujian aplikasi. Formulir pengujian aplikasi ini memiliki beberapa aspek penilaian dan pada bagian akhir terdapat permintaan saran serta kritik untuk perbaikan fitur aplikasi.

5.2.2.2 Daftar Penguji Aplikasi

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji aplikasi yang telah dibangun. Daftar nama penguji perangkat lunak ini ditunjukkan pada Tabel 5.7 untuk mahasiswa Teknik Informatika dan Tabel 5.8 untuk penguji yang bukan merupakan mahasiswa Teknik Informatika.

Tabel 5.7 Daftar Nama Penguji Mahasiswa Teknik Informatika

No.	Nama	Pekerjaan
1	Adam Widi B.	Mahasiswa
2	Irfan Hanif	Mahasiswa
3	Arij Nafi'atul Mashuda	Mahasiswa

4	Sita Nurjayanti	Mahasiswi
5	Panji Rimawan	Mahasiswa
6	Muhammad Auliaramadani	Mahasiswa

Tabel 5.8 Daftar Nama Penguji Lainnya

No.	Nama	Pekerjaan
1	Safira Nur Afifah	Mahasiswi
2	Renita Elizabeth Sianipar	
3	Puteri Nofa A.	Mahasiswi
4	Ade Nobi M.	Mahasiswa
5	Muhammad Rofiqi	Mahasiswa

5.3 Evaluasi Pengujian

Pada subbab evaluasi pengujian ini akan ditunjukkan data rekapitulasi dari hasil pengujian fungsionalitas maupun hasil pengujian pengguna yang telah dilakukan sebelumnya pada subbab 5.2. Rekapitulasi masing-masing pengujian akan dijabarkan pada subbab berikut.

5.3.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi aplikasi yang telah dipaparkan pada subbab 5.2.1 Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.9. Dari data yang terdapat pada tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

Tabel 5.9 Rekapitulasi Hasil Uji Fungsionalitas

Kode	Uji Coba		Hasil
UJ-UC-001	Uji Memilih <i>Stage</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ-UC-002	Uji Menelusuri Tempat	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
UJ-UC-003	Uji Melihat	Skenario 1	Berhasil

UJ-UC-004	Informasi	Skenario 2	Berhasil
		Skenario 3	Berhasil
	Uji Menemukan <i>Finish</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil

5.3.2 Evaluasi Pengujian Pengguna

Evaluasi pengujian pengguna dapat dilihat pada Tabel 5.10. Hasil yang terlihat merupakan hasil rata-rata dari semua kuesioner.

Tabel 5.10 Rangkuman Hasil Formulir Pengujian Aplikasi

No.	Pernyataan	Nilai
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli	4.36
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik	4.81
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)	4.81
4	Saya merasa mudah memahami aplikasi ini	4.9
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan	4.45
6	Saya merasa aplikasi sudah nyaman digunakan dan memberi pengalaman bermain yang baru	4.63

Berdasarkan Tabel 5.10, dapat diketahui nilai rata-rata adalah 4.66. Berdasarkan nilai maksimal 6, nilai ini sudah melebihi dari 4 yang memiliki arti “cukup setuju” dengan pernyataan-pernyataan yang dibuat, sehingga dapat disimpulkan bahwa *game* ini secara keseluruhan sudah cukup bagus, cukup nyaman, serta cukup mudah digunakan. Namun, masih perlu adanya pengembangan karena ini masih belum mencapai nilai bagus.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang didapatkan dari uji tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan. Selain kesimpulan, terdapat pula saran-saran untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dari proses pengerjaan tugas akhir selama tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Interaksi pengguna pada realitas virtual menggunakan Google Cardboard dapat melalui *trigger* otomatis yang berfungsi sebagai perintah “klik” dan interaksi tumbukan antara kamera dengan objek 3D.
2. Berdasarkan hasil uji coba fungsionalitas, aplikasi berhasil dibangun sesuai rancangan.
3. Berdasarkan kuesioner, responden merasa aplikasi cukup bagus, cukup nyaman dan cukup mudah digunakan.

6.2. Saran

Berikut merupakan beberapa saran untuk perbaikan dan pengembangan aplikasi dimasa yang akan datang:

1. Terdapat informasi lebih jelas di dalam aplikasi tentang bagaimana cara bermain dan bagaimana untuk menyelesaikan permainan, misalnya cerita dalam permainan.
2. Terdapat bantuan lain sehingga mempermudah pemain untuk melakukan navigasi, misalnya peta.
3. Lebih memikirkan bagaimana cara untuk mengoptimalkan aplikasi sehingga tidak mengalami patah-patah atau *lag* ketika dijalankan, terutama karena faktor grafis.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] “First-Person Shooter,” [Daring]. Tersedia pada: https://en.wikipedia.org/wiki/First-person_shooter. [Diakses 13 Desember 2016].
- [2] “Non-Player Character,” [Daring]. Tersedia pada: https://en.wikipedia.org/wiki/Non-player_character. [Diakses 13 Desember 2016].
- [3] “Virtual Reality,” [Daring]. Tersedia pada: <http://www.dictionary.com/browse/virtual--reality>. [Diakses 13 Desember 2016].
- [4] “Finite State Machines,” [Daring]. Tersedia pada: <http://www.dictionary.com/browse/finite-state-machine> . [Diakses 13 Desember 2016]
- [5] C. Murray, “State Machine,” [Daring]. Tersedia pada: <http://flylib.com/books/en/4.70.1.87/1/>. [Diakses 14 Desember 2016]
- [6] “Unity,” [Daring]. Tersedia pada: <https://unity3d.com/unity>. [Diakses 13 Desember 2016].
- [7] “Maze Game 3 on the App Store,” [Daring]. Tersedia pada: <https://itunes.apple.com/us/app/maze-game-3/id590383079?mt=8>. [Diakses 12 Desember 2017]
- [8] “Play PAC-MAN Doodle,” [Daring]. Tersedia pada: <https://google.co.id/pacman>. [Diakses 12 Desember 2017]
- [9] “Virtua Cop 2,” [Daring]. Tersedia pada: <http://www.arcadeguns.com/images/games/PC-VirtuaCop2.jpg>. [Diakses 12 Desember 2017]
- [10] “Finite State Machine,” [Daring]. Tersedia pada: <http://blog.renatopp.com/2015/03/29/finite-state->

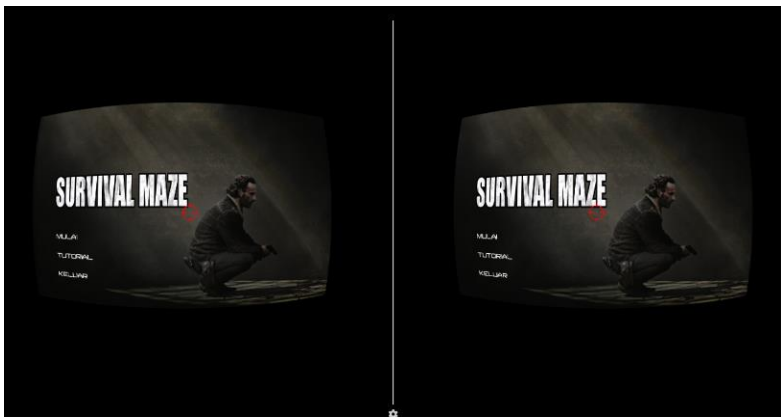
machines-in-javascript/machines-in-javascript/. [Diakses 12 Desember 2017]

- [11] “NYT VR’s First Video Shows You the Refugee Crisis Through the Eyes of Children,” [Daring]. Tersedia pada: <https://www.technobuffalo.com/2015/11/06/nyt-vrs-video-refugees-new-york-times-cardboard/>. [Diakses 12 Desember 2017]

LAMPIRAN A SCREENSHOT APLIKASI



Gambar 8.1 Tampilan awal saat membuka aplikasi



Gambar 8.2 Tampilan menu utama

Klasifikasi tumbuhan berdasar morfologi

1. **Struktur**

2. **Fungsi**

3. **Struktur**

4. **Fungsi**

5. **Struktur**

6. **Fungsi**

7. **Struktur**

8. **Fungsi**

9. **Struktur**

10. **Fungsi**

11. **Struktur**

12. **Fungsi**

13. **Struktur**

14. **Fungsi**

15. **Struktur**

16. **Fungsi**

17. **Struktur**

18. **Fungsi**

19. **Struktur**

20. **Fungsi**

21. **Struktur**

22. **Fungsi**

23. **Struktur**

24. **Fungsi**

25. **Struktur**

26. **Fungsi**

27. **Struktur**

28. **Fungsi**

29. **Struktur**

30. **Fungsi**

31. **Struktur**

32. **Fungsi**

33. **Struktur**

34. **Fungsi**

35. **Struktur**

36. **Fungsi**

37. **Struktur**

38. **Fungsi**

39. **Struktur**

40. **Fungsi**

41. **Struktur**

42. **Fungsi**

43. **Struktur**

44. **Fungsi**

45. **Struktur**

46. **Fungsi**

47. **Struktur**

48. **Fungsi**

49. **Struktur**

50. **Fungsi**

51. **Struktur**

52. **Fungsi**

53. **Struktur**

54. **Fungsi**

55. **Struktur**

56. **Fungsi**

57. **Struktur**

58. **Fungsi**

59. **Struktur**

60. **Fungsi**

61. **Struktur**

62. **Fungsi**

63. **Struktur**

64. **Fungsi**

65. **Struktur**

66. **Fungsi**

67. **Struktur**

68. **Fungsi**

69. **Struktur**

70. **Fungsi**

71. **Struktur**

72. **Fungsi**

73. **Struktur**

74. **Fungsi**

75. **Struktur**

76. **Fungsi**

77. **Struktur**

78. **Fungsi**

79. **Struktur**

80. **Fungsi**

81. **Struktur**

82. **Fungsi**

83. **Struktur**

84. **Fungsi**

85. **Struktur**

86. **Fungsi**

87. **Struktur**

88. **Fungsi**

89. **Struktur**

90. **Fungsi**

91. **Struktur**

92. **Fungsi**

93. **Struktur**

94. **Fungsi**

95. **Struktur**

96. **Fungsi**

97. **Struktur**

98. **Fungsi**

99. **Struktur**

100. **Fungsi**

101. **Struktur**

102. **Fungsi**

103. **Struktur**

104. **Fungsi**

105. **Struktur**

106. **Fungsi**

107. **Struktur**

108. **Fungsi**

109. **Struktur**

110. **Fungsi**

111. **Struktur**

112. **Fungsi**

113. **Struktur**

114. **Fungsi**

115. **Struktur**

116. **Fungsi**

117. **Struktur**

118. **Fungsi**

119. **Struktur**

120. **Fungsi**

121. **Struktur**

122. **Fungsi**

123. **Struktur**

124. **Fungsi**

125. **Struktur**

126. **Fungsi**

127. **Struktur**

128. **Fungsi**

129. **Struktur**

130. **Fungsi**

131. **Struktur**

132. **Fungsi**

133. **Struktur**

134. **Fungsi**

135. **Struktur**

136. **Fungsi**

137. **Struktur**

138. **Fungsi**

139. **Struktur**

140. **Fungsi**

141. **Struktur**

142. **Fungsi**

143. **Struktur**

144. **Fungsi**

145. **Struktur**

146. **Fungsi**

147. **Struktur**

148. **Fungsi**

149. **Struktur**

150. **Fungsi**

151. **Struktur**

152. **Fungsi**

153. **Struktur**

154. **Fungsi**

155. **Struktur**

156. **Fungsi**

157. **Struktur**

158. **Fungsi**

159. **Struktur**

160. **Fungsi**

161. **Struktur**

162. **Fungsi**

163. **Struktur**

164. **Fungsi**

165. **Struktur**

166. **Fungsi**

167. **Struktur**

168. **Fungsi**

169. **Struktur**

170. **Fungsi**

171. **Struktur**

172. **Fungsi**

173. **Struktur**

174. **Fungsi**

175. **Struktur**

176. **Fungsi**

177. **Struktur**

178. **Fungsi**

179. **Struktur**

180. **Fungsi**

181. **Struktur**

182. **Fungsi**

183. **Struktur**

184. **Fungsi**

185. **Struktur**

186. **Fungsi**

187. **Struktur**

188. **Fungsi**

189. **Struktur**

190. **Fungsi**

191. **Struktur**

192. **Fungsi**

193. **Struktur**

194. **Fungsi**

195. **Struktur**

196. **Fungsi**

197. **Struktur**

198. **Fungsi**

199. **Struktur**

200. **Fungsi**

201. **Struktur**

202. **Fungsi**

203. **Struktur**

204. **Fungsi**

205. **Struktur**

206. **Fungsi**

207. **Struktur**

208. **Fungsi**

209. **Struktur**

210. **Fungsi**

211. **Struktur**

212. **Fungsi**

213. **Struktur**

214. **Fungsi**

215. **Struktur**

216. **Fungsi**

217. **Struktur**

218. **Fungsi**

219. **Struktur**

220. **Fungsi**

221. **Struktur**

222. **Fungsi**

223. **Struktur**

224. **Fungsi**

225. **Struktur**

226. **Fungsi**

227. **Struktur**

228. **Fungsi**

229. **Struktur**

230. **Fungsi**

231. **Struktur**

232. **Fungsi**

233. **Struktur**

234. **Fungsi**

235. **Struktur**

236. **Fungsi**

237. **Struktur**

238. **Fungsi**

239. **Struktur**

240. **Fungsi**

241. **Struktur**

242. **Fungsi**

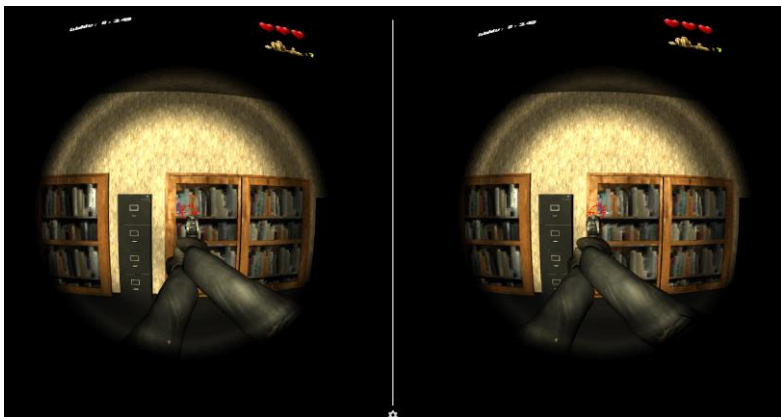
243. **Struktur**

244.

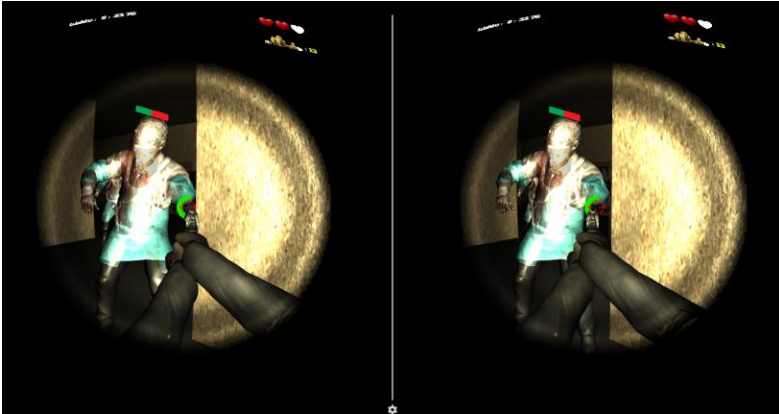
Gambar 8.4 Tampilan halaman tutorial



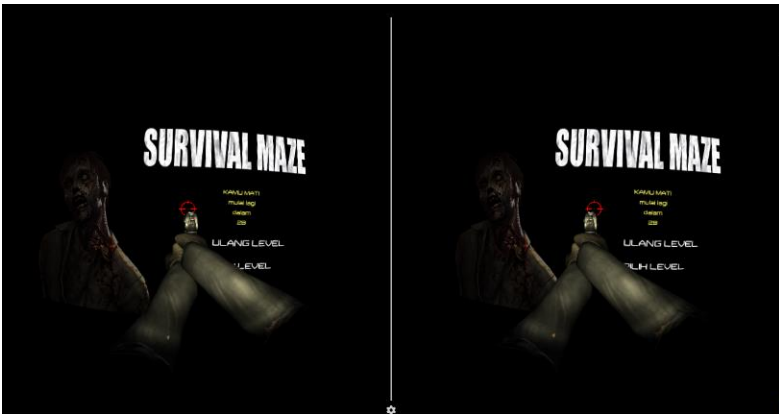
Gambar 8.5 Tampilan halaman pilih level



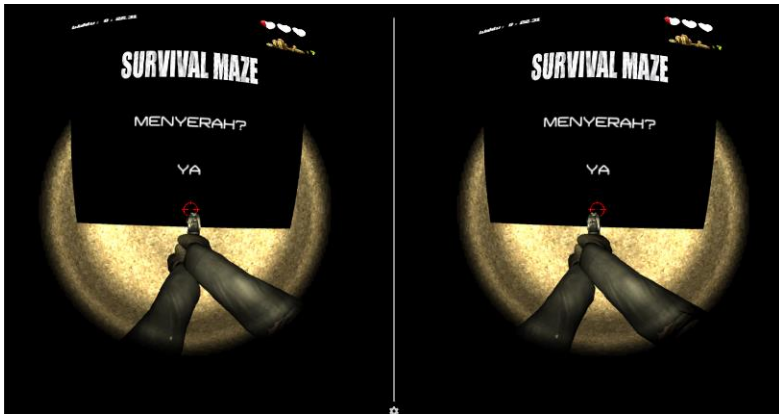
Gambar 8.6 Tampilan awal level 1



Gambar 8.7 Tampilan ketika bertemu musuh dan terjadi interaksi



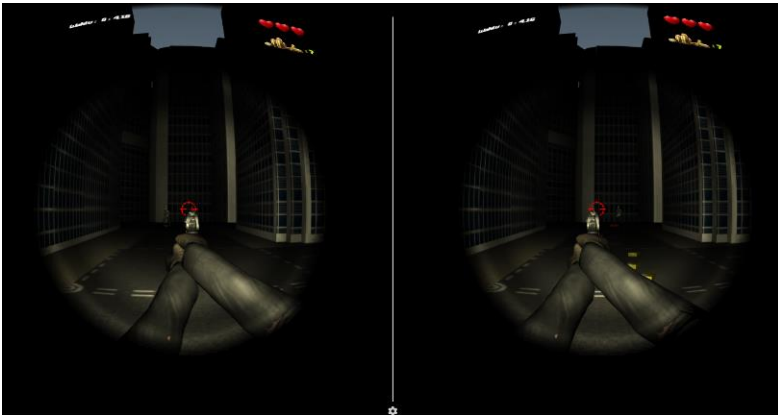
Gambar 8.8 Tampilan ketika pemain mati/*game over*



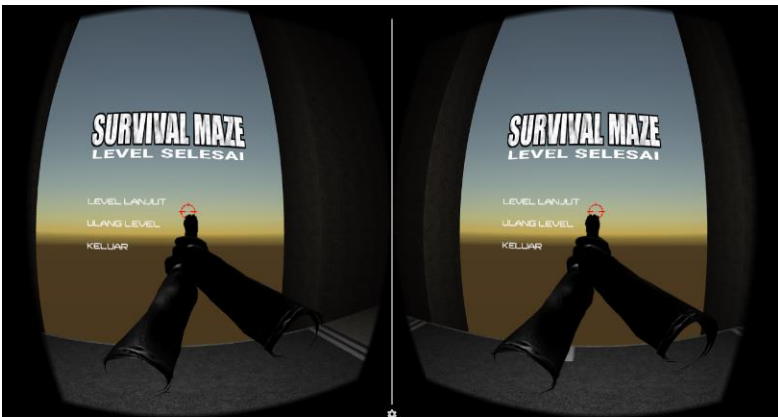
Gambar 8.9 Tampilan menu ketika pemain melihat arah atas



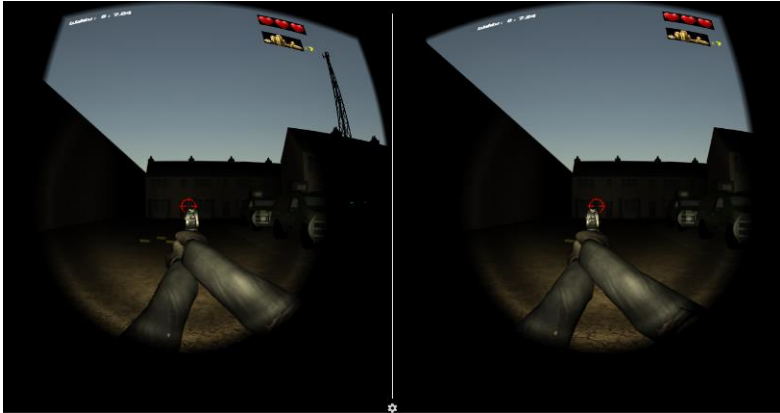
Gambar 8.10 Tampilan level 1 selesai/finish



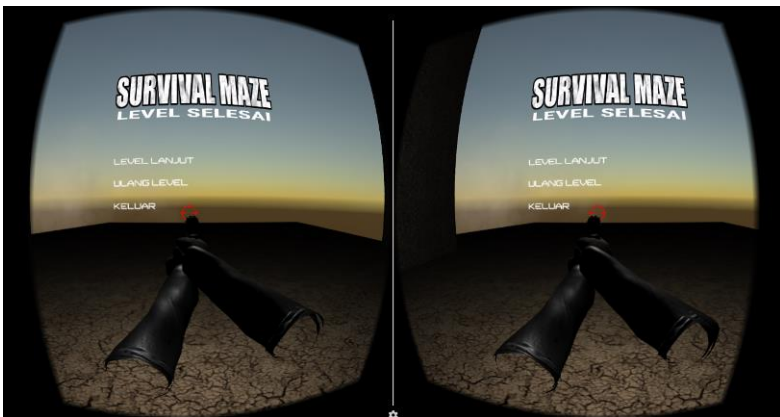
Gambar 8.11 Tampilan awal level 2



Gambar 8.12 Tampilan level 2 selesai/*finish*



Gambar 8.13 Tampilan awal level 3





Gambar 8.14 Tampilan level 3 selesai/*finish*



Gambar 8.15 Tampilan saat memilih level lanjut setelah selesai level 3 (halaman tamat)

LAMPIRAN B

KUESIONER UJI APLIKASI

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Irfan Hanif
 Umur : 21 th
 Pekerjaan : Mahasiswa (Informatika)

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	a. <input checked="" type="radio"/> Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	a. <input checked="" type="radio"/> Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	a. <input checked="" type="radio"/> Ya	b. Tidak

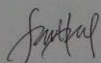
Kuisisioner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju



No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli						✓
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik						✓
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)						✓
4	Saya merasa mudah memahami aplikasi ini						✓
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan					✓	
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru						✓

Kritik dan Saran untuk pengembangan selanjutnya:
 Tambah dengan controller lain u/ jalan/pindah tempat .

Surabaya, 13 Desember 2017


 (IRFAN HANIF)

Gambar 9.1 Kuesioner Pengujian Aplikasi : Irfan Hanif

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Adam Widi B
 umur : 22
 Pekerjaan : Mahasiswa (Informatika)

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	(a.) Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	(a.) Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	(a.) Ya	b. Tidak

Kuisisioner tugas akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli				✓		
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik				✓		
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)			✓			
4	Saya merasa mudah memahami aplikasi ini					✓	
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan					✓	
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru					✓	



Kritik dan Saran untuk pengembangan selanjutnya:

Ditambah desain area bermain. Dimana warna area terlalu gelap untuk permainan VR

Surabaya, 13 Desember 2017

(Adam Widi B)

Gambar 9.2 Kuesioner Pengujian Aplikasi : Adam Widi B.

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Arij Nafi'atul Mashuda
 Umur : 20
 Pekerjaan : Mahasiswa (Informatica)

Silahkan lingkari (O) pada pilihan yang sesuai

1. Apakah Anda mengetahui tentang realitas virtual?	a. Ya	b. Tidak
2. Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	a. Ya	b. Tidak
3. Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	a. Ya	b. Tidak

Kuisisioner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli					✓	
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik					✓	
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)					✓	
4	Saya merasa mudah memahami aplikasi ini					✓	
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan						✓
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru					✓	

Kritik dan Saran untuk pengembangan selanjutnya:

Screennya dibuat tidak ngelur



.....

.....

Surabaya, 12 Desember 2017

Arij N M

Gambar 9.3 Kuesioner Pengujian Aplikasi : Arij Nafi'atul Mashuda

Kuisloner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Sita Nurjayanti
 Umur : 21
 Pekerjaan : Mahasiswa (Informatika)

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	(a) Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	(a) Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	(a) Ya	b. Tidak

Kuisloner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4 : Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli					✓	
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik				✓		
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)						✓
4	Saya merasa mudah memahami aplikasi ini						✓
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan					✓	
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru					✓	

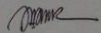
Kritik dan Saran untuk pengembangan selanjutnya:

Gamarnya sudah bagus, tapi kurang suka dengan objek musuh-musuh (horror).



.....

.....

Surabaya, 12 - 12 2017


 (Sita Nurjayanti)

Gambar 9.4 Kuesioner Pengujian Aplikasi : Sita Nurjayanti

Kuisloner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Panji Rimawan
 Umur : 22
 Pekerjaan : Mahasiswa (Informatika ITS)

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	<input checked="" type="radio"/> a. Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	<input checked="" type="radio"/> a. Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	<input checked="" type="radio"/> a. Ya	b. Tidak

Kuisloner tugas akhir:


Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli					✓	
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik					✓	
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)				✓		
4	Saya merasa mudah memahami aplikasi ini					✓	
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan				✓		
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru					✓	



Kritik dan Saran untuk pengembangan selanjutnya:

Optimisasi grafik agar permainan lancar dijalankan (tidak lag) dan bisa dijalankan mullus di semua perangkat yg mendukung VR baik yang spek tinggi ataupun rendah sekalipun.

Surabaya, 12-12.....2017

()

Gambar 9.5 Kuesioner Pengujian Aplikasi : Panji Rimawan

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Muhammad Auliaramadani
 Umur : 19
 Pekerjaan : Mahasiswa (Informatika)

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	<input checked="" type="radio"/> a. Ya	<input type="radio"/> b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	<input checked="" type="radio"/> a. Ya	<input type="radio"/> b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	<input checked="" type="radio"/> a. Ya	<input type="radio"/> b. Tidak

Kuisisioner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli				<input checked="" type="checkbox"/>		
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik				<input checked="" type="checkbox"/>		
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)			<input checked="" type="checkbox"/>			
4	Saya merasa mudah memahami aplikasi ini				<input checked="" type="checkbox"/>		
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan				<input checked="" type="checkbox"/>		
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru			<input checked="" type="checkbox"/>			

Kritik dan Saran untuk pengembangan selanjutnya:

• Untuk meningkatkan "game play" meningkatkan jalannya permainan



• Untuk menambah fitur AI yang tidak mengganggu jalannya permainan

• Tambahkan story dan aktivitas untuk player supaya pemain lebih asyik dalam bermain

Surabaya, 12 Desember 2017

Kus

Gambar 9.6 Kuesioner Pengujian Aplikasi : Muhammad Auliaramadani

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : SAFIRA NUR AFIFAH
 Umur : 21 TH
 Pekerjaan : MAHASISWA

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	<input checked="" type="radio"/> a. Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	<input checked="" type="radio"/> a. Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	<input checked="" type="radio"/> a. Ya	b. Tidak

Kuisisioner Tugas Akhir:


Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli				✓		
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik						✓
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)						✓
4	Saya merasa mudah memahami aplikasi ini						✓
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan				✓		
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru					✓	



Kritik dan Saran untuk pengembangan selanjutnya:

- Level 1 terlalu sulit untuk dimainkan
 - sudah cacat untuk dikoreksi.

Surabaya, 12 Des 2017


 (SAFIRA NUR AFIFAH)

Gambar 9.7 Kuesioner Pengujian Aplikasi : Safira Nur Afifah

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Renita Elizabeth Sianipar

Umur : 22

Pekerjaan :

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	<input checked="" type="radio"/> a. Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	<input checked="" type="radio"/> a. Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	<input checked="" type="radio"/> a. Ya	b. Tidak

Kuisisioner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai

1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju


No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli			✓			
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik				✓		
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)					✓	
4	Saya merasa mudah memahami aplikasi ini					✓	
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan					✓	
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru				✓		

Kritik dan Saran untuk pengembangan selanjutnya:

Kritik: objek kurang jelas untuk pengguna yang memiliki mata minus. Pencahuran fokus VR Box kurang baik.



Interface saat bermain kurang jelas (keterangan sisa peluru tidak terlihat)

Surabaya, 12 Desember 2017

()

Renita E.S.

Gambar 9.8 Kuesioner Pengujian Aplikasi : Renita Elizabeth Sianipar

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Anaroid Survival Maze"

Nama : PUTERI NOFA -A
 Umur : 22
 Pekerjaan : Mahasiswa

Silahkan lingkari (a) pada pilihan yang sesuai

1 Apakah Anda mengetahui tentang realitas virtual?	a. Ya	b. Tidak
2 Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	a. Ya	b. Tidak
3 Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	a. Ya	b. Tidak

Kuisisioner tugas akhir:

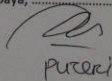
Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli			✓			
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik					✓	
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)					✓	
4	Saya merasa mudah memahami aplikasi ini					✓	
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan			✓			
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru				✓		



Kritik dan Saran untuk pengembangan selanjutnya:

Sound sama bagian menjerah ophonnya kurang tsadest!
Semuaanya apik, zombienya pulak.

Surabaya, 12 Desember 2017


 Puteri

Gambar 9.9 Kuesioner Pengujian Aplikasi : Puteri Nofa A.

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : Ade Nobi M
 Umur : 21
 Pekerjaan : Mahasiswa

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	a. Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	a. Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	a. Ya	b. Tidak

Kuisisioner tugas akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli				✓		
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik					✓	
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)					✓	
4	Saya merasa mudah memahami aplikasi ini				✓		
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan				✓		
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru				✓		



Kritik dan Saran untuk pengembangan selanjutnya:

Bagus dapat pengalaman baru dalam bermain. Tapi lebih baik ada peta dalam gamenya meskipun tidak ditunjukkan di suatu area tersebut ada zombie atau lainnya. Cara pantulan senam seperti jalan kelan. Patahkan bukan.

Surabaya, 12 Desember 2017

Ade Nobi M

Gambar 9.10 Kuesioner Pengujian Aplikasi : Ade Nobi M.

Kuisisioner Tugas Akhir "Permainan First Person Shooter (FPS) Realitas Virtual Perangkat Android Survival Maze"

Nama : MUHAMMAD ROFIQI
 Umur : 21
 Pekerjaan : MAHASISWA

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah Anda mengetahui tentang realitas virtual?	<input checked="" type="radio"/> a. Ya	b. Tidak
2	Apakah Anda pernah memainkan aplikasi bergenre First Person Shooter (FPS), misalnya Doom, Counter Strike dan lain-lain?	<input checked="" type="radio"/> a. Ya	b. Tidak
3	Apakah Anda pernah memainkan aplikasi bergenre labirin sebelumnya?	<input checked="" type="radio"/> a. Ya	b. Tidak

Kuisisioner Tugas Akhir:

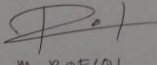
Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4: Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Pernyataan	Nilai					
		1	2	3	4	5	6
1	Saya merasa objek-objek dalam permainan sudah terlihat nyata seperti asli					<input checked="" type="checkbox"/>	
2	Saya merasa aplikasi memiliki tampilan dan desain yang menarik					<input checked="" type="checkbox"/>	
3	Saya merasa nyaman melihat aplikasi (model/objek tidak patah-patah)					<input checked="" type="checkbox"/>	
4	Saya merasa mudah memahami aplikasi ini			<input checked="" type="checkbox"/>			
5	Saya merasa tombol-tombol kontrol pada permainan tidak membingungkan				<input checked="" type="checkbox"/>		
6	Saya merasa aplikasi sudah nyaman digunakan dan memberikan pengalaman bermain yang baru					<input checked="" type="checkbox"/>	

Kritik dan Saran untuk pengembangan selanjutnya:

Menyemangkan dan bikin kayak saya bingung ini misinya harus ngapain terus harus kemana

Surabaya, 12 Desember 2017


 (M. ROFIQI)

Gambar 9.11 Kuesioner Pengujian Aplikasi : Muhammad Rofiqi

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis lahir di kota Magelang pada tanggal 17 Januari 1995. Dalam perjalanan hidupnya penulis pernah menempuh pendidikan dasar di SDN Cacaban 3 Magelang, dilanjutkan pendidikan menengah SMPN 1 Magelang, tingkat atas di SMAN 1 Magelang dan terakhir sedang menempuh pendidikan di perguruan tinggi negeri Institut Teknologi Sepuluh Nopember Surabaya (ITS Surabaya) jurusan Departemen Informatika.

Selama menjadi mahasiswa penulis pernah mengikuti pelatihan pengembangan diri mahasiswa dari tingkat Pra-Dasar, Dasar dan beberapa pelatihan lainnya. Penulis aktif dalam UKM Paduan Suara Mahasiswa (PSM) ITS hingga mengikuti beberapa Big Event bersama UKM tersebut, penulis juga pernah berkecimpung di beberapa Big Event lainnya diantaranya Schematics ITS 2014 dan 2015. Untuk bidang organisasi, penulis pernah tergabung menjadi pengurus Himpunan Mahasiswa Teknik Computer-Informatika (HMTIC) ITS 2014/2015 serta pengurus UKM PSM ITS 2014/2015 dan 2015/2016. Selain itu penulis juga pernah melaksanakan Kerja Praktik di PT Miconos Yogyakarta. Penulis dapat dihubungi melalui email guruharyasenna@gmail.com